VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

AUTOMATICKÉ NAVRHOVÁNÍ KLÍČOVÝCH SLOV

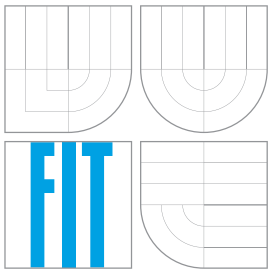
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. TOMÁŠ STRACHOTA

BRNO 2010

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

AUTOMATICKÉ NAVRHOVÁNÍ KLÍČOVÝCH SLOV

AUTOMATIC KEYWORD SUGGESTION

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. TOMÁŠ STRACHOTA

VEDOUCÍ PRÁCE
SUPERVISOR

doc. RNDr. PAVEL SMRŽ, Ph.D.

BRNO 2010

Abstrakt

Práce mapuje teoretický základ pro vytvoření systému pro automatické navrhování klíčových slov. Obsahuje přehled současných statistických metod pro vyhledávání termínů a metod hodnocení vyhledávání. Na základě těchto známých přístupů navrhuje možná vylepšení vyhledávání. Byla zkoumána možnost sjednocování klíčových slov podle synonym, předúprava vstupních textů a úprava slov do výsledných tvarů.

Abstract

This thesis surveys theoretical background for automatic keyword suggestion system. It contains overview of current statistical term recognition methods and methods for evaluation of automatic term recognition systems. Based on the known approach the thesis specifies possible enhancements. It explores unifying keywords using thesauri, input text filtering and correction of word forms.

Klíčová slova

automatické rozpoznání termínů, rozpoznávání klíčových slov, měření úspěšnosti rozpoznávání

Keywords

automatic term recognition, keyword recognition, recognition evaluating

Citace

Tomáš Strachota: Automatické navrhování klíčových slov, diplomová práce, Brno, FIT VUT v Brně, 2010

Automatické navrhování klíčových slov

Prohlášení

Prohlašuji, že jsem tento semestrální projekt vypracoval samostatně pod vedením pana doc. RNDr. Pavla Smrže, Ph.D.

.....
Tomáš Strachota
26. května 2010

Poděkování

Především děkuji vedoucímu diplomové práce doc. RNDr. Pavlu Smržovi, Ph.D. za odborné vedení a poskytnutí cenných informací. Dále děkuji Tomáši Lokajovi za jeho ochotu při zprostředkování testovacích dat.

Obsah

Kapitola 1

Úvod

Klíčovým slovem v lingvistice rozumíme výraz, který se váže na nějaký text a je pro něj určitým způsobem významný. Měl by výstižně shrnovat podstatu obsahu tohoto textu. Proto můžou mít mezi klíčovými slovy místo jak obecnější výrazy, které vystihují text jako celek, tak přesné termíny, kterých se téma textu dotýká. Volba několika málo vhodných klíčových slov může být mnohdy obtížným úkolem i pro člověka, což vede k zamyšlení nad automatizací tohoto procesu. Zároveň ale už tato prvotní myšlenka naznačuje složitost takového úkolu pro stroj.

Dříve sloužila v oblasti informačních technologií klíčová slova především k vyhledávání v dokumentech. Ke každému dokumentu bylo přiřazeno několik slov, které jej charakterizovaly a na základě kterých potom mohl být vyhledán. S nástupem fulltextového vyhledávání začal tento způsob použití mizet. Stále však existují oblasti, kde najdou klíčová slova upotřebení. Mohou posloužit například k automatické kategorizaci dokumentů podle obsahu, k tvorbě rejstříků publikací, nebo například pro vyhledání termínů v textech. Každé z jmenovaných použití má pochopitelně svá specifika. V žádném z případů se však nejedná o triviální problém. Zkušenosti s metodami pro vyhledávání klíčových slov se mohou použít také v širším kontextu zpracování přirozeného jazyka.

Cílem předkládané diplomové práce je prozkoumat možnosti současných metod pro získávání termínů z textu a na jejich základě vytvořit systém pro automatické navrhování klíčových slov. Výsledný systém by měl být dostatečně univerzální, aby mohl být využit pro vyhledávání v různých jazycích. V rámci této práce bude vytvořena podpora pro práci s českými a anglickými texty. Kvalita vytvořených nástrojů bude měřena pro oba jazyky na dostupných dokumentech tak, aby vynikla kvalita jednotlivých částí.

Na začátku práce nejprve shrnuji teoretický úvod do vyhledávání klíčových slov. Kapitola 2 se zabývá známými statistickými metodami pro vyhledávání termínů. Kapitola 3 uvádí přehled metod měření úspěšnosti takového vyhledávání. V následující kapitole 4 je uveden návrh systému se zamýšlenými metodami vylepšení vyhledávání, vzniklý na základě nasbíraných poznatků. Podrobnosti o implementaci návrhu uvádím v kapitole 5, na kterou navazuje kapitola 6 o konfiguraci vytvořeného systému a dosažených výsledcích. V závěru shrnuji přínos práce jako celku.

Přestože jsem se v předkládané práci snažil používat českou terminologii, vyskytují se v textu i anglická slova. Jedná se o případy, kdy neexistuje dostatečně zaužívaný český ekvivalent. Překlad by mohl způsobit nejednoznačnost a proto bylo nutné slevit ze stylistických požadavků. Stejně tak názvy všech algoritmů jsou záměrně uvedeny v jejich původním znění, aby nemohlo dojít k záměnám při překladu.

Kapitola 2

Statistické algoritmy pro vyhledáván termínů

V oblasti *automatického vyhledávání termínů*[1] se obvykle postupuje ve dvou krocích [13]. Nejdříve jsou lingvistickým filtrem vybrána kandidátní slova. Tito kandidáti jsou následně ohodnoceni statistickými metodami a jsou z nich určeni ti s nejlepším hodnocením.

Podle přístupu k výpočtu ohodnocení se dělí statistické algoritmy do dvou kategorií [14][33]:

- *termhood metody* měří příslušnost termínu k dané doméně. Tyto metody jsou založeny především na měření frekvencí výskytu slov. Je možné je použít jak pro jednoduché (jednoslovné), tak pro komplexní (víceslovné) termíny.
- *unithood metody* měří sílu spojení slov v termínu. Určují tedy, zda se spolu jednotlivé složky termínu vyskytují pouze náhodou, nebo tvoří kolokaci. Jsou použitelné jen pro komplexní termíny.

V následující kapitole představíme 9 statistických metod, které byly vybrány buď na základě jejich dobrých výsledků v jiných studiích, nebo pro svoji jednoduchost, protože se jedná o metody vyjadřující základní přístup k problematice.

2.1 Termhood metody

2.1.1 Term frequency

Term frequency vyjadřuje množství všech výskytů kandidátního klíčového slova v textu. Předpokládá se, že slova s častějším výskytem jsou důležitější. *Term frequency* $Tf(i)$ se spočítá jako normalizovaná frekvence slova i v množině dokumentů K.

$$Tf(i) = \frac{f(i)}{\sum_k f(k)} \tag{2.1}$$

Tato jednoduchá metoda je často využívána v lingvistickém předzpracování pro prvotní seřazení kandidátů.

[1] Anglicky *Automatic Term Recognition*, často zkracováno jen jako ATR.

2.1.2 Term frequency - Inverse document frequency

Tato metoda zohledňuje fakt, že významná slova se vyskytují v určitém dokumentu často, ale v ostatních dokumentech (pojednávajících o jiné tematice) už může být jejich výskyt řidší. *Inverse document frequency* $Idf(i)$ počítá výskyty termínu i v jednotlivých dokumentech a vyjadřuje tak jeho důležitost v kolekci dokumentů D:

$$Idf(i) = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|} \tag{2.2}$$

Výsledná hodnota $Tf(i)Idf(i)$ je pak součinem:

$$Tf(i)Idf(i) = Tf(i).Idf(i) \tag{2.3}$$

V případě, že je metoda použita na kolekci dokumentů z jedné domény a výstupem použití této metody má být seznam termínů pro celou kolekci spíše než výběr termínů pro každý dokument zvlášť, je možné výpočet $Tf(i)Idf(i)$ upravit. Frekvenci $Tf(i)$ můžeme za těchto okolností vyjádřit jako počet výskytů slova w_i v dokumentech kolekce. Výpočet frekvence se tedy provádí jakoby v kolekci byl jenom jeden doménově specifický dokument.

2.1.3 Term frequency - Inverse paragraph frequency

Term frequency - Inverse paragraph frequency je obměnou předchozí metody. Místo počítání výskytu termínů v různých dokumentech zohledňuje rozdíl jejich výskytů v odstavcích. Myšlenka tohoto algoritmu vychází z úvahy, že zkoumané dokumenty jsou členěny do logických sekcí, které jsou tematicky zaměřeny. Předpokladem pro funkčnost algoritmu je správné rozdělení dokumentů do odstavců. Hodnocení termínu se vypočte jako

$$Tf(i)Idf(i) = Tf(i).Ipf(i), \tag{2.4}$$

kde $Ipf(i)$ je

$$Ipf(i) = \log \frac{|P|}{|\{p_j : t_i \in p_j\}|}, \tag{2.5}$$

přičemž P je množina odstavců zkoumaného dokumentu.

2.1.4 Residual inverse document frequency

Residual inverse document frequency je jistou alternativou k *Term frequency - Inverse document frequency*. Upřednostňuje ty termíny, jejichž počet výskytů je vyšší, než je pravděpodobné. $RIDF(i)$ je definováno jako rozdíl logaritmů skutečné frekvence a frekvence předpovězené pomocí Poissonova rozdělení.

$$RIDF(i) = Idf(i) - \log(1 - p(0; \lambda(i))) \tag{2.6}$$

Ve vzorci je p Poissonovo rozdělení s parametrem $\lambda(i) = \frac{f(i)}{D}$, který vyjadřuje průměrný počet výskytů v dokumentu. $1 - p(0; \lambda(i))$ je Poissonova pravděpodobnost, že se v dokumentu termín i vyskytuje aspoň jednou.

Poissonova pravděpodobnost se vypočte jako

$$p(k; \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}, \tag{2.7}$$

z čehož po dosazení získáme výsledný vzorec pro výpočet Residual inverse document frequency

$$RIDF(i) = Idf(i) - \log(e^{-\frac{f(i)}{D}}) \tag{2.8}$$

2.1.5 Domain consensus

Domain consensus je jedna z metod navržených pro nástroj TermExtractor [32]. Vychází z myšlenky, že termíny musí být zaužívané a měly by se tudíž vyskytovat v textech napříč celou doménou. Jeho hodnota se spočítá jako

$$DC(t) = \sum_{d_i \in D} P(t, d_i) \log(P(t, d_i)), \tag{2.9}$$

kde $P(t, d_i)$ je složená distribuce pravděpodobnosti, která může být aproximována jako

$$E(P(t, d_i)) = \frac{f(t, d_i)}{\sum_{d_j \in D} f(t, d_j)}, \tag{2.10}$$

přičemž $f(t, d_j)$ je frekvence termínu t v dokumentu d_j.

Domain consensus nabývá vysokých hodnot, pokud je pravděpodobnostní rozdělení termínů v dokumentech rovnoměrné.

2.1.6 Weirdness

Tato metoda je založena na úvaze, že rozdělení termínů ve specializovaném korpusu se bude významně lišit od rozdělení v obecném korpusu. Tento předpoklad je vyjádřen vzorcem

$$Weirdness(i) = \frac{\frac{f_s(i)}{n_s}}{\frac{f_g(i)}{n_g}} \tag{2.11}$$

kde $f_s(i)$ je frekvence výskytů slova ve specializovaném (doménovém) korpusu a n_s celkový počet slov tohoto korpusu. Obdobně $f_g(i)$ a n_g vyjadřují stejné veličiny pro obecný korpus.

Metoda *Weirdness* byla původně vytvořena pro ohodnocení jednoslovných termínů. Pro práci s víceslovnými termíny počítáme geometrický střed hodnot *Weirdness* pro jednotlivá slova.

2.1.7 Likelihood ratio

Likelihood ratio, která je popsána v [19], vychází ze stejných předpokladů, jako *Weirdness*. K měření rozdílu mezi frekvencemi výskytů slov v doménovém a obecném korpusu se však používá statistického testu. První hypotéza je, že pravděpodobnost výskytu daného slova v naší doméně je stejná, jako pravděpodobnost výskytu v background modelu. Druhá hypotéza je, že pravděpodobnost výskytu slova v doméně je významně vyšší, než v background korpusu. Pro frekvence výskytů slova předpokládáme binomické rozdělení.

$$p = \frac{f_s + f_g}{n_s + n_g} \qquad p_s = \frac{f_s}{n_s} \qquad p_g = \frac{f_g}{n_g}$$

f opět značí frekvenci slova ve specializovaném (f_s) a v obecném korpusu (f_g). Stejně tak n_s, n_g jsou počty všech slov v korpusu.

$$LR = \log L(f_s, n_s, p) + \log L(f_g, n_g, p) - \log L(f_s, n_s, p_s) - \log L(f_g, n_g, p_g) \tag{2.12}$$

$$L(n, k, x) = x^k (1 - x)^{n-k} \tag{2.13}$$

2.1.8 BM25

BM25 [2] je metodou pro vyhledávače, určenou k ohodnocení dokumentů na základě slov z vyhledávacího dotazu [29]. Stejně dobře ale může sloužit k ohodnocení termínů, pokud se na problém podíváme z opačné strany. Z textu budeme vybírat kandidátní klíčová slova, která použijeme jako vyhledávací dotaz. Metoda vrátí ohodnocení, které vyjadřuje vhodnost dokumentu k danému dotazu, což zároveň vyjadřuje vhodnost dotazu pro daný dokument. Systematickým ohodnocením všech kandidátů jsme tak schopni vybrat vhodná klíčová slova.

V průběhu let byla rovnice pro výpočet *BM25* upravována a mírně pozměňována. Princip však zůstává stejný. Uveďme základní tvar vzorce, stejně jako jej uvádí Wikipedie [1].

Budiž Q dotaz obsahující klíčová slova $q_1, \ldots, q_n$, pak hodnocení metodou BM25 se vypočítá jako

$$score(D,Q) = \sum_{i=1}^{n} IDF(q_i) \cdot \frac{f(q_i,D) \cdot (k_1+1)}{f(q_i,D) + k_1 \cdot (1 - b + b \cdot \frac{|D|}{avgdl})}, \tag{2.14}$$

kde $f(q_i, D)$ je frekvence slova q_i v dokumentu D, $|D|$ je délka dokumentu D měřená ve slovech a $avgdl$ je průměrná délka dokumentů v množině. Proměnné b a k_1 jsou volné parametry obvykle volené jako $k_1 = 2.0$ a $b = 0.75$, přičemž b je nutné volit z intervalu $b \in \langle 0, 1 \rangle$.

IDF je *Inverse document frequency*, která se pro případ *BM25* spočítá jako

$$IDF(q_i) = \log \frac{N - n(q_i) + 0.5}{n(q_i) + 0.5}, \tag{2.15}$$

kde N je celkový počet dokumentů v korpusu a $n(q_i)$ je počet dokumentů, které obsahují q_i.

2.2 Unithood metody

2.2.1 Lexical cohesion

Metoda *Lexical cohesion* je stejně jako *Domain consensus* použita v TermExtractoru [32]. Původně však byla uvedena v [26], kde se v tehdejší literatuře oproti ostatním metodám měřící kohezi ukázala jako efektivnější. Snaží se poměrem výskytu termínu jako celku a výskytu jeho jednotlivých slov vyjádřit, jak moc spolu tato slova souvisejí.

Nechť $n = |t|$ je počet slov, ze kterých se skládá termín $t = w_1 w_2 w_3 ... w_n$. *Lexical cohesion* je pak měřena jako:

$$LC(t) = \frac{n.f(t).\log f(t)}{\sum_{j=0}^{n} f(w_j)} \tag{2.16}$$

2.2.2 C-Value

C-Value je metoda navržená speciálně pro výběr komplexních termínů [6]. Vzorec je postaven na třech základních principech:

- výběr nejfrekventovanějších termínů

[2] Známá také jako *Okapi BM25* podle vyhledávacího frameworku v němž byla použita.

- penalizace vnořených termínů, které se objeví jako podřetězec delšího kandidátního termínu
- délka termínu vyjádřená počtem slov

Vzorec pro výpočet *C-Value* je následující

$$C - value(a) = \begin{cases} \log_2|a| \cdot f(a) & \text{pokud je } a \text{ vnořené} \\ \log_2|a| \cdot (f(a) - \frac{1}{|T_a|} \sum_{b \in T_a} f(b)) & \text{jinak} \end{cases} \tag{2.17}$$

kde a je kandidátní termín, $f()$ je frekvence výskytu termínu v korpusu, T_a je množina kandidátních termínů které obsahují a jako podřetězec a $P(T_a)$ je počet takových termínů.

2.3 Kombinování statistických metod

Často je výhodné použít kombinaci několika metod najednou. Přínos kombinování *termhood* a *unithood* přístupu byl ověřen například v [27], kde byly výstupy jednotlivých algoritmů zkombinovány pomocí *AdaBoost* algoritmu [7].

Jiný přístup k současnému použití více metod byl zkoumán v [35]. V této práci bylo využito volícího algoritmu s váhami pro jednotlivé algoritmy. Výsledné ohodnocení termínu bylo vypočteno jako

$$rank = \sum_{i}^{k} \frac{1}{R(t_i)} w_i, \tag{2.18}$$

kde k je počet algoritmů, které se kombinují, $R(t_i)$ je ohodnocení termínu t vypočtené algoritmem i a w_i je váha pro tento algoritmus. w_i je odhadnuto jako

$$w_i = \frac{P_i}{\sum_i^k P_i}, \tag{2.19}$$

kde P_i je *přesnost* algortimu i. Pokusy ukázaly, že takto kombinované ohodnocení dává lepší výsledky než samostatně použité algoritmy. Dobré výsledky tohoto přístupu byly potvrzeny i v [13].

Kapitola 3

Měření úspěšnosti vyhledávání

Při tvorbě systémů pro extrakci termínů, klíčových slov, či rejstříků z textu vyvstává otázka, jakým způsobem je hodnotit. Narážíme na několik problémů.

Zaprvé není jednoduché stanovit referenční výstup, se kterým se mají výsledky porovnávat. Konkrétně u klíčových slov může být hodnocení, zda se jedná o vhodné klíčové slovo, nebo ne, značně subjektivní. Při hodnocení se tedy musíme vypořádat s faktem, že i slovo, které není v referenční množině, nemusí být nutně nevhodné.

Porovnání nijak nezjednodušuje ani sama podstata extrahovaných dat. Slova mohou být jednak v různých tvarech a jednak je ne zřídkakdy možné použít synonym.

Další problematickou částí jsou víceslovné termíny. Někdy i termín kratší o jedno slovo může mít stále dostatečnou vypovídací hodnotu, přestože se v referenční množině nenachází.

V následující kapitole se těchto problematických oblastí dotkneme a nastíníme možná řešení.

3.1 Testovací a trénovací data

Základním předpokladem pro správné měření výsledků vyhledávacího systému je oddělení trénovací a testovací množiny dat. Testovací množinu není možné používat v průběhu tvorby systému. Pokud by došlo k tomu, že stejná data se použijí jak pro trénování tak pro testy, budou výsledky testů znehodnoceny.

Pro testování extraktoru klíčových slov budou použity dva zdroje dat. Prvním z nich je sada norem ČNI. Obsahuje asi 100 norem z nejrůznějších technologických odvětví. Konkrétně budou využity části s terminologií, které obsahují vždy termín a jeho vysvětlení. Velkou výhodou norem je, že text v nich bývá vícejazyčný, zarovnaný podle jednotlivých termínů. To umožňuje poměrně spolehlivé porovnání extraktoru pro české a pro anglické texty. Jako referenční klíčová slova budou brány názvy termínů.

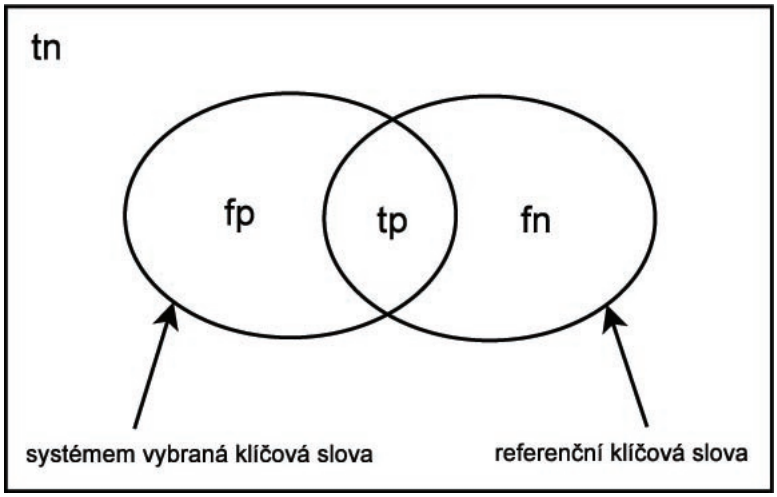
Normy byly k dispozici ve formátu MS Word. Poloautomatickým zpracováním za pomoci vytvořených Visual Basic skriptů byla z norem extrahována terminologie, která byla roztříděna podle jazykových verzí a následně uložena jako čistý text. Výsledkem je cca 1.5 MB textu pro každou jazykovou verzi.

Další sadou testovacích dat jsou anglické články z konferencí, které jsou k dispozici pro testování nejrůznějších projektů z oblasti zpracování přirozeného jazyka na fakultě informačních technologií VUT v Brně. Některé články obsahují přímo klíčová slova vyjmenovaná v úvodu. Tato se dají využít jako referenční klíčová slova vůči kterým bude porovnáván výstup nástroje vytvořeného v rámci této práce. Souhrnná velikost všech článků je cca 41 GB.

3.2 Metody měření úspěšnosti

3.2.1 Přesnost a úplnost

Nejčastěji používanou metodou pro měření úspěšnosti vyhledávacích systémů je použití veličin *přesnost* a *úplnost*. Ilustrujme na příkladu z oblasti extrahování klíčových slov.



Obrázek 3.1: Diagram znázorňující přesnost a úplnost.

V diagramu jsou znázorněny čtyři množiny:

- tp - *true positives* jsou klíčová slova, která systém vybral správně
- tn - *true negatives* jsou slova o kterých systém správně rozhodl, že nejsou důležitá a nezařadil je mezi klíčová
- fp - *false positives* (často také chyby 2. typu) jsou klíčová slova, která systém vybral, avšak neměla být vybrána
- fn - *false negatives* (také chyby 1. typu) jsou klíčová slova, která systém nevybral, avšak měla být vybrána

Přesnost je definována jako poměr správně vybraných klíčových slov vůči všem vybraným. Vyjadřuje, jaké množství balastu je extrahováno.

$$pesnost = \frac{tp}{tp + fp} \tag{3.1}$$

Úplnost je definována jako poměr správně vybraných klíčových slov vůči referenčním. Vyjadřuje tak, kolik klíčových slov systém opomenul.

$$plnost = \frac{tp}{tp + fn} \tag{3.2}$$

3.2.2 F-measure

F-measure kombinuje *přesnost* a *úplnost* do jediné veličiny [19]. Vypočítá se jako

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}}, \tag{3.3}$$

kde P je *přesnost*, R je *úplnost* a α je váha určující poměr mezi P a R. Často se volí $\alpha = 0.5$, což má za následek rovnocenný vliv P a R na výslednou hodnotu. Rovnici je v takovém případě možné zjednodušit na

$$F = \frac{2PR}{R+P}. \tag{3.4}$$

3.2.3 Zohlednění uspořádání výsledků

Předchozí dvě metody ukazovaly pouze hodnocení výsledné množiny bez ohledu na její uspořádání. Na následujícím příkladu si však ukážeme, že i mezi systémy, které by tímto způsobem byly hodnoceny stejně, se můžou skrývat výrazné rozdíly ve výkonu. Příklad vychází z [19].

Uvažujme 3 systémy, které ohodnocují 6 klíčových slov k1 až k6, přičemž polovina z nich je vhodných. Všechny systémy mají shodnou *přesnost* 0.5. Přesto je zřejmě systém 1 lepší než systém 3 a ten je lepší než systém 2.

hodnocení	systém 1	systém 2	systém 3
	k1✓	k5	k4
	k2✓	k4	k1✓
	k3✓	k6	k2✓
	k4	k2✓	k5
	k5	k1✓	k3✓
	k6	k3✓	k6
přesnost na úrovni 3	1	0	0.66
přesnost na úrovni 6	0.5	0.5	0.5
neinterpolovaná průměrná přesnost	1	0.38	0.58

Tabulka 3.1: Příklad různě hodnotících systémů s jejich mírami úspěchu.

Metod zohledňujících uspořádání výsledků je několik. Nejjednodušší z nich je měření na určitých úrovních počtu vybraných položek. V našem příkladě na úrovni 3 a 6. Z výsledků pak vystoupí postupné změny hodnot sledovaných veličin.

Další možností je *neinterpolovaná průměrná přesnost*. Ta na rozdíl od předchozí metody shrnuje výsledek do jednoho čísla. Vypočítá se jako průměr hodnot *přesnost* pro každý bod výsledného seznamu, ve kterém se nachází správně vybraná položka.

Naopak *interpolovaná průměrná přesnost* vychází z hodnot *přesnost* vypočítaných pro určité předem dané úrovně hodnot *úplnost*. Obvykle 0 % až 100 % s krokem 10. Hodnoty jsou počítány postupně a pokud dojde ke zvýšení *přesnost*, je tato hodnota zvýšena zpětně všem předchozím bodům, které mají hodnotu nižší. Tím dochází k interpolaci. Tento postup je založen na myšlence, že pokud přesnost roste, budeme se chtít podívat na více výsledků.

Vzhledem k tomu, že mezi *přesnost* a *úplnost* existuje v praxi závislost (*přesnost* obvykle klesá se stoupající hodnotou *úplnost*), může být přínosné vynést graf přesnosti v závislosti na úplnosti.

Pro jednoduché porovnání výsledků může postačit i výpočet průměrného umístění správně vybraných termínů. Systém, který pracuje lépe, umístí více správných termínů na začátku seznamu, což má za následek nižší průměrnou hodnotu. Tato metoda nemá velkou vypovídací hodnotu o celkových kvalitách systému, ale pro porovnávání postačí.

3.3 Zpřesnění měření

Předchozí metody pro měření úspěchu striktně porovnávaly výstup s referenčním výstupem a předpokládaly, že klíčové slovo je vybráno dobře, nebo špatně. V praxi však narážíme na fakt, že rozhodování nemusí být tak jednoznačné a to především u víceslovných klíčových slov. Budeme demonstrovat, jaké problémy můžou vzniknout a navrhneme úpravy, které umožní podrobněji ukázat kvality a nedostatky systému.

3.3.1 Tvary slov

Především u českých textů je nutné vybraná klíčová slova převést do základních tvarů. Tento výsledný tvar může být správně určený, ale odlišný od tvaru v referenční množině. Přestože druhý případ nastává velmi zřídka, může k němu dojít. Například existují slova, která mají více spisovných tvarů. Klíčová slova se také mohou lišit v čísle. Konkrétní případy ukazuje tabulka 4.1.

vybrané kl. slovo	referenční kl. slovo
Newtonův zákon	Newtonovy zákony
literatura	literatúra
tankový kanón	tankový kanon

Tabulka 3.2: Ukázka kl. slov ve správných tvarech, které jsou však odlišné od tvarů referenčních kl. slov.

Řešení je poměrně snadné. Stačí převést slovo po slově do základního tvaru a porovnávat takto vzniklá lemmata. Tento test oddělí chyby úpravy do výsledných tvarů od chyb zbytku systému.

3.3.2 Pořadí slov

Často dochází k tomu, že pořadí slov ve víceslovných výrazech z referenční množiny se liší od pořadí slov ve výrazu vybraném extraktorem, byť význam obou je stejný. Jako příklad vezměme klíčové slovo, které se skládá z podstatného jména a několika přídavných jmen. Je-li referenční množina tvořena z rejstříku, je pravděpodobné, že přídavná jména budou v termínu až za podstatným jménem. Toto pořadí slov je doporučováno i v normě [25]. Naopak v běžném textu se spíše přikloníme k variantě, kdy přídavná jména předchází jméno podstatné. Konkrétní příklady jsou pro názornost v tabulce 3.3.

Nabízí se jednoduché řešení. Aby byla mezi správné výsledky započtena i klíčová slova s jiným pořadím slov, je potřeba při testování pro každé klíčové slovo vytvořit množinu jeho permutací a zjišťovat, jestli se některý prvek z této množiny nachází v množině referenční. Ukázka takové množiny permutací je pro doplnění v tabulce 3.4. Tabulka slouží pouze pro demonstraci řešení problému a v praxi by se permutace pochopitelně generovaly z lemmat jednotlivých slov.

vybrané kl. slovo	referenční kl. slovo
říční koryto	koryto říční
souvislá vysoká oblačnost	vysoká oblačnost souvislá
řídká vysoká oblačnost	vysoká oblačnost řídká

Tabulka 3.3: Ukázka vybraných kl. slov a jejich protějšků s jiným pořadím slov v referenční množině.

původní kl. slovo	výrazy v jeho množině permutací
souvislá vysoká oblačnost	souvislá vysoká oblačnost
	souvislá oblačnost vysoká
	vysoká souvislá oblačnost
	vysoká oblačnost souvislá
	oblačnost souvislá vysoká
	oblačnost vysoká souvislá

Tabulka 3.4: Množina permutací pro výraz „souvislá vysoká oblačnost“.

Vytváření permutací je však vhodné jen u určité kombinace slovních druhů. Pokud bychom přehodili pořadí dvou podstatných jmen, narazíme na problém. Výrazy *barva lahve* a *lahev barvy* očividně nemají stejný význam. Proto je nezbytné generování permutací omezit jen na kombinace slovních druhů, u nichž nedojde při přehození pořadí ke změně smyslu celého výrazu.

3.3.3 Synonyma

Dalším problémem, který může nastat při porovnávání výsledků, je výskyt synonym. Někteří autoři ve snaze předejít opakování stejných slov v textech střídají použití synonym. S velkou pravděpodobností bude v referenční množině pouze jedno z těchto možných synonym. Pokud bude jako klíčové slovo vybráno jiné synonymum, vyhodnotí se pochopitelně chybně jako nesprávné. Příklady takových chyb jsou uvedeny v tabulce 3.5.

vybrané kl. slovo	referenční kl. slovo
acidní výluh	kyselý výluh
akcelerace	zrychlení
psí plemeno	psí rasa
rádius otáčení	poloměr otáčení

Tabulka 3.5: Ukázka vybraných kl. slov a jejich synonym v referenční množině.

Částečným řešením je použít slovník synonym a kontrolovat všechny možné kombinace. Dosažený výsledek bude pochopitelně záviset na kvalitě použitého slovníku.

3.3.4 Nedokonalá klíčová slova

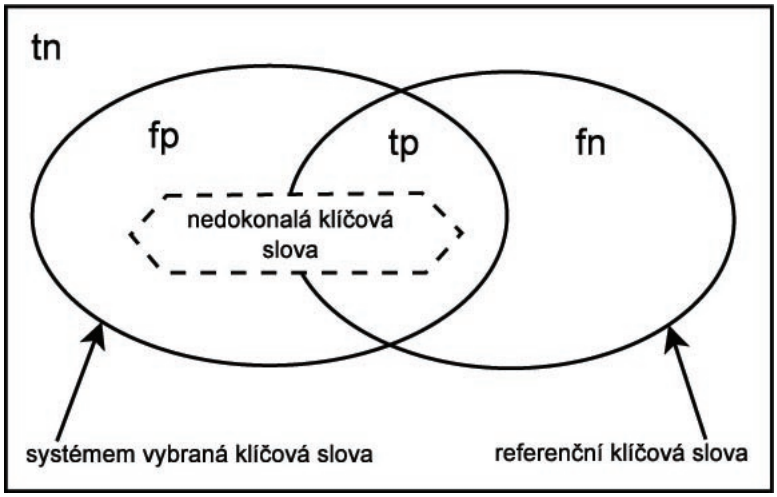
Velmi často se stává, že systém vybere delší klíčové slovo, než je požadováno. Typicky se objevuje o jedno až dvě přídavná jména před vlastním kl. slovem navíc. Jako příklad

uvažujme následující větu:

Pokud na hmotný bod nepůsobí žádné síly, označujeme jej jako volný hmotný bod.

Požadovaným klíčovým slovem z této věty by bylo zřejmě sousloví *hmotný bod*. Na poli klíčových slov by ale nebylo velkou chybou, pokud by systém vybral slova *volný hmotný bod*, protože část textu se tímto pojmem pravděpodobně bude zaobírat.

V [15] bylo navrženo řešit tento problém pomocí takzvaných *nedokonalých klíčových slov*. Jedná se o klíčová slova, která obsahují referenční klíčové slovo jako podřetězec. I *nedokonalá klíčová slova* byla započítávána jako správně vybraná.



Obrázek 3.2: Diagram znázorňující přesnost a úplnost s nedokonalými klíčovými slovy.

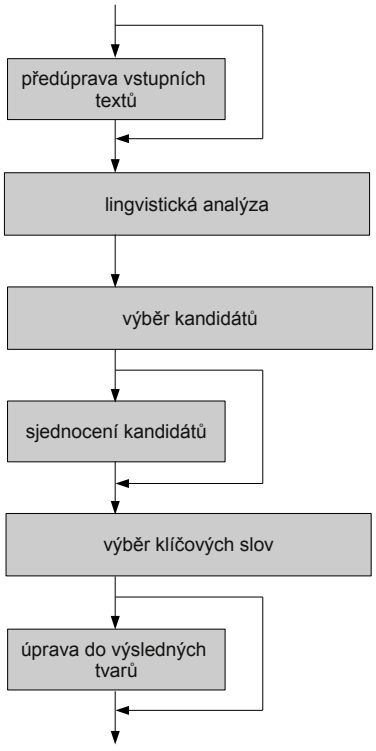
V téže práci [15] se však ukázalo, že tato metoda příliš vylepšuje výsledky testů. A to až do té míry, že se stávají nepoužitelnými. Jedinou spolehlivou možností je ruční kontrola. Je však nemyslitelné, aby byly výsledky každého testu procházeny ručně a kontrolovány slovo po slově. Proto je vhodné jen v opodstatněných případech projít část výstupu a určit procentuální odhad kolik klíčových slov je vhodných, přestože se nenachází v referenční množině.

Kapitola 4

Navržený systém

Na základě studia současného stavu extrakce klíčových slov a termínů z jiných prací a na základě vlastních zkušeností z bakalářské práce bylo identifikováno několik požadavků na můj vlastní extraktor. Kromě klasických přístupů, které se již dříve osvědčily, má systém za cíl prozkoumat i ne zcela tradiční pohledy na věc.

Už v návrhu byl systém rozdělen na šest na sobě relativně nezávislých částí, při čemž od některých se očekává pouze zkvalitnění finálního výstupu a je možno je vypustit. Takové rozdělení by mělo zjednodušit případné budoucí modifikace. Návaznost jednotlivých částí je znázorněna v diagramu běhu systému 4.1.



Obrázek 4.1: Diagram návaznosti součástí systému.

Na vstup systému přicházejí textové dokumenty, v nichž mají být vyhledána klíčová slova. Před jakýmkoliv lingvistickým zpracováním souborů je vhodné upravit jejich obsah

a odstranit ty části, které by mohly negativně ovlivnit další práci s textem. Každý soubor pak projde morfologickou analýzou, která jej rozdělí na jednotlivé věty a určí ke každému slovu jeho druh. Z takto označkovaného obsahu bude následně vybrána široká množina kandidátů na klíčová slova. Tuto množinu je možno dále upravovat a rozšiřovat, což dává prostor k vyhledávání klíčových slov, která se nevyskytují přímo ve vstupním textu, avšak vyplývají nepřímo z jeho obsahu. Kandidáty je dále třeba ohodnotit nějakou mírou, která bude udávat jejich kvalitu. Nejlépe ohodnocené výrazy se stanou klíčovými slovy, které systém pošle na výstup. Ještě před tím je možné upravit tvary klíčových slov, což je důležité zejména u víceslovných výrazů, kde nestačí ponechat všechna slova v prvním pádu.

Jelikož je systém navrhován od základů, nemusíme se nechat příliš omezovat konceptem již existujících knihoven, či cizích nástrojů. Návrh je možné udělat dostatečně univerzální a veškerou návaznost na vstupy z vnějšku odstínit. Vzhledem k tomuto faktu bude navržen framework, který umožní práci s více jazyky a integraci různých externích nástrojů. Na takovém frameworku pak bude možné stavět další složitější programy.

V následujících částech kapitoly podrobně rozeberu návrh každé části systému v pořadí, v jakém se podílí na extrakci (viz obr. 4.1). Jedinou výjimkou bude podkapitola Předúprava vstupních textů. Vzhledem k tomu, že je možné předúpravu zcela vypustit a celou svojí podstatou stojí jakoby mimo zbytek systému, bude uvedena jako poslední.

4.1 Lingvistická analýza

Nástroje lingvistické analýzy dokáží provést automatický rozbor textu. Obvykle tak činí na základě statistických údajů nasbíraných z korpusů příslušného jazyka. Úrovně, na kterých je rozbor prováděn, se liší nástroj od nástroje. Pro předzpracování textu se v zásadě používají tři úrovně analýzy:

- *rozdělení slov* - nejjednodušší úroveň zpracování, určuje hranice slov a vět. Používá se hlavně jako základní zpracování vstupu před analýzou na vyšší úrovni.
- *určení mluvnických kategorií* - určí pro každé slovo věty nejčastěji jeho lemma a slovní druh, případě i jiné mluvnické kategorie. Tato úroveň je pro předzpracování nejpoužívanější, protože dokáže poskytnout dostačující množství informací v obvykle rozumně krátkém čase.
- *určení závislostních stromů* - nad každou větou vytvoří její závislostní strom. Pro automatické zpracování většinou méně využívaná úroveň rozboru, neboť pro většinu úkolů je zbytečně složitá a postačí předchozí úroveň.

I pro vyhledávání klíčových slov by byla postačující druhá z uvedených úrovní zpracování. Ke každému slovu je nutné znát především jeho slovní druh a lemma. Slovní druh se využívá při vyhledávání vhodných kandidátů na klíčová slova. Lemma se zužitkuje při počítání četností výskytů slov, aby mohla být i pro různé tvary téhož slova identifikována příslušnost k tomuto slovu. To je obzvlášť důležité při zpracovávání tak vysoce flektivních jazyků, jako je čeština.

V následujících odstavcích představím tři nástroje pro lingvistickou analýzu a shrnu možnosti, které nabízejí. O využití analýzy a přínosu závislostních stromů pro vyhledávání klíčových slov se potom zmíním blíže v části 4.2.

4.1.1 Pražský závislostní korpus

Pražský závislostní korpus, neboli Prague Dependency Treebank (dále jen PDT), je projekt pro ruční anotaci velkého množství českých textů lingvistickou informací. Motivací ke vzniku projektu byly myšlenky zformulované v článku [8]. Ve své první verzi obsahovala pouze anotaci morfologie a povrchové syntaxe. Verze PDT 2.0 přidává navíc hloubkovou syntax a sémantiku, aktuální členění, koreferenci a lexikální sémantiku založenou na valenčním slovníku [10].

PDT vznikla za účelem aplikovat teoretické výsledky Pražské lingvistické školy a pomocí metod strojového učení vytvořit spolehlivé nástroje automatické analýzy a generování jazykových dat. Především druhý úkol vyžaduje velké množství zpracovaných vět z přirozených textů. PDT 2.0 obsahuje cca 2 milióny slov s provázanými anotacemi na úrovni morfologie, z čehož 1,5 miliónu je anotováno také na úrovni povrchové syntaxe a zhruba 0,8 miliónu na úrovni hloubkové syntaxe a sémantiky.

Kromě nástrojů pro prohlížení korpusu a anotovaných vět vznikly v rámci PDT 2.0 také nástroje pro analýzu textů. Ty umožňují zpracovávat data na čtyřech rovinách:

- *slovní rovina* - rozděluje text do dokumentů a odstavců. Jsou tu rozlišeny slovní jednotky (slova, čísla, interpunkce) a jsou opatřeny jednoznačnými identifikátory
- *morfologická rovina* - má za úkol rozdělit text na věty a ke každému slovu určit jeho mluvnické kategorie (neboli značku) a lemma. Přichází-li pro daný tvar slova v úvahu více možností, uvede každou z nich.
- *analytická rovina* - navazuje na morfologickou rovinu a z každé věty vytvoří orientovaný strom s kořenem, s ohodnocenými hranami a uzly. Každý prvek morfologické roviny odpovídá právě jednomu uzlu. Hrany stromu vyjadřují závislost mezi slovními jednotkami. Typ vztahu je dán funkčním ohodnocením hrany. Na této rovině už jsou značky a lemmata jednoznačné.
- *tektogramatická rovina* - i zde jsou věty reprezentovány orientovanými stromy s kořenem, s ohodnocenými hranami a uzly. Strom zachycuje hloubkovou strukturu věty. Uzly zastupují pouze plnovýznamová slova a jsou obohaceny o typ kontextového zapojení. Navíc jsou do stromu přidávány uzly, které neodpovídají žádnému prvku z morfologické vrstvy (např. uzly pro nevyjádřený podmět apod.).

4.1.2 TreeTagger

Jak už název napovídá, TreeTagger [31] je nástroj pro určování slovních druhů a lemmat v textu. V prvotní verzi byl určen pro angličtinu, avšak po natrénování na příslušném korpusu může být použit i pro jiné jazyky. K dispozici jsou slovníky a konfigurační soubory například pro němčinu, španělštinu a italštinu.

Podobně jako značkovací systémy, které používají Markovovy modely [5][12], odhaduje TreeTagger pravděpodobnost výskytu sekvencí různých slovních druhů a přiklání se k nejpravděpodobnější variantě. K určení pravděpodobností však TreeTagger využívá binárních rozhodovacích stromů. Strom se buduje upraveným algoritmem ID3 na trénovací množině n-gramů. Výsledný strom je poté ještě ořezán, aby neobsahoval uzly, které přinášejí minimální zisk.

TreeTagger pracuje se slovníkem, podle kterého určuje pro příslušné slovo pravděpodobnosti jednotlivých značek. Slovník se skládá ze dvou částí: slovník úplných výrazů a

slovník koncovek. Ty jsou postupně prohledávány v pořadí, jak jsou uvedeny. Slovník úplných výrazů byl v anglické verzi vytvořen z korpusu Penn Treebank [20] (použito cca 2 mil. slov). Jestliže v něm není slovo nalezeno, TreeTagger se pokusí vyhledat koncovku v druhém slovníku a přiřadit pravděpodobnosti podle ní. Slovník koncovek byl vytvořen ze statistik nasbíraných o otevřených slovních druzích, protože jen u těch se dá předpokládat výskyt neznámých slov a tím pádem smysluplnost koncovkové analýzy. Pokud ani tady TreeTagger neuspěje, přiřadí slovu implicitní hodnoty.

Testy ukázaly, že TreeTagger je schopný pro anglické texty dosáhnout přesnosti cca 95% s drobnými rozdíly podle mocnosti n-gramů, které byly použity pro trénování [31]. To se dá, vezmeme-li v potaz rychlost, jakou je program schopný zpracovávat dokumenty, považovat za velmi dobrý výsledek.

4.1.3 MiniPar

MiniPar [18] je nástroj pro automatickou tvorbu závislostních stromů nad větami. Každému slovu přiřadí lemma, slovní druh a přidělí mu jeho místo v rámci zapojení do stromu. MiniPar vznikl jako přímý potomek programu PrinciPar [17]. Snahou při tvorbě MiniParu bylo převzít myšlenky minimalistického programu a vytvořit tak vysoce efektivní nástroj.

Gramatiku, kterou MiniPar potřebuje k vytváření stromů, reprezentuje jako síť, v níž uzly představují gramatické kategorie a hrany závislostní vztahy mezi nimi. Síť obsahuje 35 uzlů a 59 hran, přičemž další prvky jsou přidávány dynamicky a slouží k reprezentaci podkategorií sloves.

Analýza věty probíhá ve třech krocích. Nejprve je použita lexikální analýza, která rozdělí vstupní text na základní lexikální jednotky. Pomocí algoritmu *Message passing* [16] je následně vytvořen sdílený les větných stromů. V poslední fázi je na základě ohodnocení vybrán nejlepší strom.

Slovník MiniParu byl odvozen z projektu WordNet [23] a obsahuje zhruba 130 000 záznamů. V každém záznamu je uložen seznam možných slovních druhů pro dané slovo. Rozhodování lexikální víceznačnosti má na starosti až parser, který tvoří větné stromy.

Testy na americkém korpusu SUSANNE [30] ukázaly, že MiniPar je schopen dosáhnout přesnost 89% a úplnosti 79% [18].

4.2 Výběr kandidátních klíčových slov

Tato část extraktoru má za úkol připravit množinu výrazů, které budou dále zkoumány a zbytek systému určí, zda se jedná o klíčová slova, nebo ne. Kdyby nebyl text předzpracován lingvistickou analýzou, museli bychom se spolehnout na jednoduché vybírání n-tic slov o délce v určitém rozsahu. To by však mohlo způsobit, že i n-tice s nulovou informační hodnotou by mohly být zbytkem systému například na základě četností jejich výskytu vyhodnoceny jako klíčové slovo. Proto je nanejvýš vhodné využít znalosti mluvnických kategorií slov určených předchozí částí systému a vybírat si jen určité posloupnosti slov.

Zaměříme-li se na slovní druhy, ze kterých se klíčová slova skládají, zjistíme, že se opakuje několik nejčastějších. Kromě tradičního dělení slovních druhů (kterých máme v češtině 10), můžeme z lingvistického hlediska dělit slova také na syntaktické a sémantické slovní druhy [22]. Sémantické slovní druhy rozlišujeme čtyři:

- sémantická substantiva (vyjadřují substanci)
- sémantická adjektiva (vyjadřují vlastnost)

- sémantická adverbia (vyjadřují okolnost)
- sémantická slovesa (vyjadřují událost)

Dále rozlišujeme autosémantické slovní druhy, u kterých se sémantický slovní druh shoduje s tradičním slovním druhem. Právě autosémantická slova nesou největší informaci a vyskytují se v klíčových slovech nejčastěji.

Při výběru kandidátů v označkovaném textu máme několik možností, jak specifikovat, které posloupnosti slovních druhů se mají vybrat. Základním postupem je přímo vyjmenovat sekvence slovních druhů. O něco pokročilejší je specifikace regulárních výrazů, kterým musí posloupnost značek vyhovovat. To přináší tu výhodu, že se nemusíme omezovat na předem dané délky n-tic. Další úpravou může být vyhledávání n-tic s dírami. To umožní najít i výrazy, jejichž slova se nevyskytují v textu přímo vedle sebe. Příklad takového výrazu najdeme v následujících dvou větách [21].

- *Volný vstup* vám nemůžeme zaručit.
- *Vstup* na výstavu je *volný* pouze v pondělí.

Nebezpečí tohoto přístupu je, že bude nadgenerovávat velké množství n-tic slov, které spolu ve skutečnosti nijak nesouvisejí.

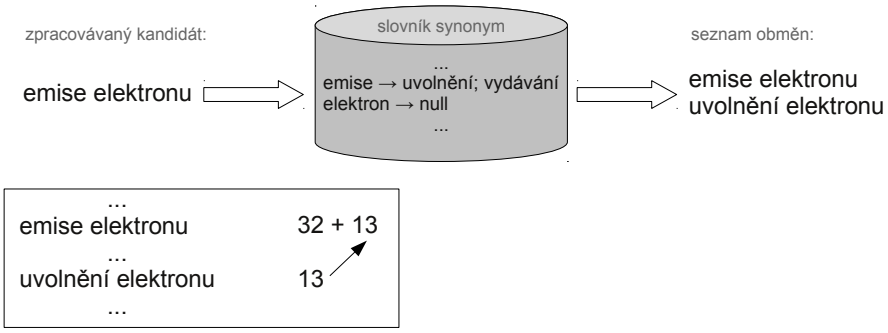
Pokud jsou po lingvistické analýze k dispozici závislostní stromy vět, můžeme je s výhodou použít. Závislost jednotlivých slov na sobě už je vyjádřena stromem a proto není třeba uvažovat díry v souslovích. Vyhledávání vhodných n-tic se potom realizuje pomocí hledání podstromů, které vyhovují specifikovaným pravidlům. Tento přístup byl použit i pro potřeby extraktoru, kterým se zaobírá tato práce.

4.3 Sjednocení klíčových slov

Většina autorů textů se snaží vyhnout se opakovanému použití stejných slov v krátkém úseku textu. Vyhledáváme-li odborné výrazy, pak nás tento fakt neohrozí, protože v nich většinou nelze změnit žádné slovo, aniž by se vytratil původní význam či přesnost výrazu. U klíčových slov je ale situace jiná. Klíčová slova obvykle vystihují větší celek textu a mohou mít i obecnější charakter. Proto se může stát, že ve snaze docílit stylisticky pěknějšího textu použije autor dokumentu synonymum slova, které je určitým způsobem pro dokument významné. Vzhledem k tomu, že většina metod pro určení klíčových slov vychází z počítání četností výskytu slova, bude jejich výsledek nepříznivě ovlivněn, protože četnost se rozptýlí mezi více slov, jejichž význam je ovšem stejný. Dokonce je možné aby došlo k situaci, kdy se klíčové slovo v textu vůbec nenachází. Potom by při použití klasických metod pro toto slovo selhalo úplně.

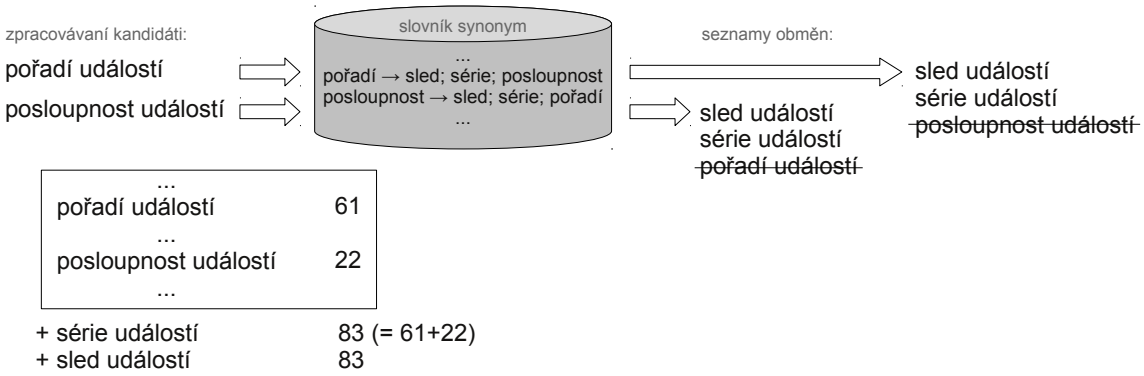
Pokud by byl k dispozici dostatečně kvalitní slovník synonym, je možné pomocí něj tento problém řešit. Pro každé klíčové slovo by pak bylo možné vytvořit pomocí synonym jeho obměny. Dále by byly v množině kandidátů vyhledány obměněné verze a původnímu výrazu by se zvýšila četnost výskytů o hodnotu četnosti všech nalezených obměn. Demonstrujme postup na příkladu pro výraz *emise elektronu* (viz obr. 4.2). Jeho původní četnost je 32. Na základě slovníku synonym byly vytvořeny obměny *uvolnění elektronu* a *vydávání elektronu*. Výraz *uvolnění elektronu* se v množině kandidátů vyskytuje s četností 13, výraz *vydávání elektronu* nalezen nebyl. Zvýšíme proto původnímu kandidátu četnost o 13.

Problém s nalezením klíčových slov, která se v textu nevyskytují vůbec, je možné řešit obdobně. Nejprve by se vypočítaly obměny pro všechny kandidáty v množině. Následně



Obrázek 4.2: Demonstrace postupu zvyšování četnosti podle slovníku synonym pro kandidáta *uvolnění elektronu*.

by byla množina obohacena o ty obměny, které byly vytvořeny pro více než jednoho kandidáta. Jejich četnost by byla vypočtena jako součet četností všech původních výrazů, které obměna pokrývá. Opět demonstrujme ukázkou (viz obr. 4.3). V množině kandidátů se nachází výrazy *posloupnost událostí* a *pořadí událostí*. Podle záznamů v synonymickém slovníku vytvoříme obměny obou výrazů a vybereme ty, které mezi kandidáty ještě nejsou. Tento postup má i svou negativní stránku. Pokud by se totiž ve slovníku vyskytovalo pro slovo *událost* například synonymum *příhoda*, dojde k zařazení nových obměn, jako je *pořadí příhod* a podobné. Takový výraz je však nepřesný a může mít odlišný význam.



Obrázek 4.3: Demonstrace postupu vytváření nových kandidátů podle slovníku synonym.

Problémem nadále zůstává nalezení dostatečně kvalitního slovníku synonym. Požadavky na synonyma se navíc mohou lišit podle domény v níž je hodláme použít. Při tvorbě extraktoru pro tuto práci byl použit slovník WordNet [23][24] a anglický i český thesaurus z projektu OpenOffice.org [2].

WordNet je rozsáhlá databáze anglického jazyka, která vznikla na půdě univerzity v Princetonu. Do takzvaných synsetů, které sdružují slova stejného nebo podobného významu, ukládá podstatná jména, přídavná jména, příslovce a slovesa. Jednotlivé synsety jsou provázány sémantickými a lexikálními vztahy. Typy vztahů se liší podle slovních druhů, které synset shlukuje. U podstatných jmen jsou to například synsety s hypernymy[1], hypo-

[1] Slova významově nadřazená.

nymy[2], holonymy[3] a meronymy[4]. Celkově databáze WordNetu obsahuje asi 152 000 slov.

Méně kvalitní jsou slovníky používané v opensource kancelářském balíku OpenOffice.org. Ty se omezují jen na ukládání synonym ve tvaru slovo a k němu příslušný seznam slov stejného významu. Anglický slovník byl vytvořen na základě již zmiňované databáze WordNet a má obdobný rozsah. Jiný formát uložení a odstranění některých informací umožňuje oproti původní databázi rychlejší zpracování. Slovník zachovává členění podle slovních druhů, což pomůže zpřesnit vyhledání synonym. Český synonymický slovník pochází z půdy Fakulty informačních technologií Masarykovy Univerzity. Zahrnuje jen 23 000 slov, z čehož můžeme domýšlet, že se zřejmě kvalitou nebude svým anglickým protějškům rovnat.

4.4 Výběr klíčových slov

Ze seznamu kandidátů je nutné vybrat užší množinu, kterou můžeme s určitou pravděpodobností prohlásit za klíčová slova. To se obvykle provádí ohodnocením kandidátů pomocí statistických metod a výběrem určitého počtu nejlépe hodnocených. Tyto metody již byly představeny v kapitole 2. Výsledky jednotlivých algoritmů je možné kombinovat a spojit tak dobré vlastnosti více z nich. Návrh počítá s univerzálním řešením, které dokáže kombinovat libovolné množství hodnotících algoritmů. Zároveň musí být možné měnit strategie pro kombinaci ohodnocení. Kromě samotného hodnocení je nutné se vypořádat také s tím, jakým způsobem se budou ukládat data, která jsou pro výpočty potřebná.

Mnoho hodnotících algoritmů využívá srovnávání frekvence výskytu slova v textu s jeho četností v obecném korpusu. Takové referenčním četnosti jsou pochopitelně z důvodu rychlého přístupu k nim předpočítány předem. Obvykle se jedná o poměrně velké soubory, což způsobuje problémy s jejich načtením do paměti. To vede k zamyšlení, jak je uložit efektivně.

Jako první se nabízí omezit nějakým způsobem rozsah vlastního souboru četností. Pro potřeby vyhledávání klíčových slov je vhodné neukládat četnost každého tvaru slova, ale pouze jejich lemmat. Tím se zmenší množství ukládaného textu a zároveň se zpřesní výpočty s frekvencemi. Dalším krokem je neukládat veškeré slovní druhy. Pokud je dopředu známo, že jako kandidáti budou vybrány například jen kombinace přídavných a podstatných jmen, je zbytečné ukládat jiná slova, protože jejich frekvence nás nebude nikdy zajímat. Posledním krokem je odstranění četností menších, než je určitá mez. Tento postup se však už nedá považovat za zcela čistý, protože zasahujeme do souboru hodnot a ovlivňujeme ho.

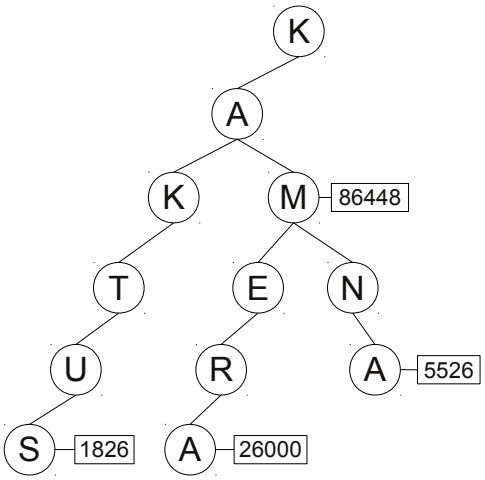
Paměťové úspory je možno dosáhnout i samotným způsobem uložení. Jednou z vhodných struktur je *trie* [4]. Jedná se o strom, který využívá faktu, že velké množství slov spolu sdílí několik počátečních písmen a je zbytečné je ukládat znova. Proto každý uzel stromu obsahuje informaci o písmeni, které reprezentuje, seznam dceřiných uzlů a případně informaci o četnosti slova, které v uzlu končí. Způsob ukládání ilustruje obrázek 4.4. Pro účely uložení slovníků se dají použít také minimální automaty, které kompresi ještě zdokonalují.

Některé statistické metody vyžadují pro správný výpočet nejenom referenční četnosti jednotlivých slov, ale také četnosti celých kandidátů. To znamená, že buď musíme mít uloženy také četnosti n-tic, nebo je musíme být schopni aproximovat. Pro odhad četností víceslovných výrazů se často používá výpočet průměru z hodnot pro jednotlivá slova, což může výsledek zkreslit. Vhodnější je jako aproximaci používat nejnižší z frekvencí výskytů

[2] Slova významově podřazená.

[3] Slovo X je holonymem slova Y, pokud Y je součást X (např. dům je holonymum ke slovu okno).

[4] Slovo X je meronymem slova Y, pokud X je součást Y. Jedná se o opačný vztah k holonymu.



Obrázek 4.4: Ilustrace ukládání četností slov do struktury trie. Uložena slova *kaktus*, *kam*, *kamera* a *kamna*.

jednotlivých slov, protože je jisté, že pokud bychom nasbírali data o výskytech n-tic ze stejného korpusu, mohla by se n-tice objevit maximálně stejně často, jako její jednotlivá slova. V této práci zvolím kompromis a budou předpočítány četnosti jak jednotlivých slov, tak jejich dvojic. Hodnoty pro delší n-tice budou odhadovány již zmíněným způsobem. Slova v souboru četností budou podle potřeby filtrována tak, jak bylo popsáno výše.

4.5 Úprava slov do výsledných tvarů

Až do této chvíle byla klíčová slova, se kterými jsme pracovali, z výše uvedených důvodů v jejich základním tvaru, který určila analýza. Tento tvar však není, alespoň co se českého jazyka týče, vhodný pro finální výstup jako výsledné klíčové slovo.

Základní tvar (lemma) slova se určuje jako první pád jednotného čísla. U přídavných jmen a příslovcí je lemma v prvním stupni. Přídavná jména navíc přebírají rod od řídicího slova podle pravidel mluvnické shody [11]. To však není při určení základního tvaru k dispozici, proto se užívá mužský rod. U sloves se jako lemma používá infinitiv. Ten je, jako jediný možné přímo použít jako výstup. Pokud je klíčovým slovem jediné podstatné jméno, mohli bychom se spokojit se základním tvarem také. Za zvážení však stojí, jestli jako výsledný tvar vybrat jednotné nebo množné číslo, protože v určitých situacích může být vhodnější jedno, nebo druhé. Daleko složitější je ale určení finálního tvaru u víceslovných klíčových slov. U těch musíme vzít v potaz nejenom číslo, ale především rod a pád závislých slov. Tabulka 4.1 ukazuje o jak netriviální úkol se jedná. Například ne všechny předložky se totiž pojí jen s jedním pádem, čímž vzniká nejednoznačnost, se kterou si musíme při volbě výsledného tvaru poradit.

Pokud máme k dispozici závislostní stromy vět, ve kterých se klíčová slova objevila, je možné pokusit se základní tvar vyhledat přímo v textu. Tím obejdeme peripetie s určováním správných mluvnických kategorií a spolehneme se na inteligenci autora textu. Klíčová slova tvoří v původních větách podstromy. Je třeba vyhledat takový podstrom, jehož kořen (řídicí slovo) je v prvním pádě. Na výstup pak převezmeme celý původní tvar tohoto výskytu klíčového slova včetně jeho čísla.

základní tvar	slova ve vhodném tvaru
let na Měsíc	let na Měsíc
pobyt na Měsíc	pobyt na Měsíci
plazmatický membrána	plazmatická membrána
koryto řeka	koryto řeky

Tabulka 4.1: Víceslovná klíčová slova v základních tvarech a jejich příslušné správně vytvořené výsledné tvary.

Jestliže není vhodný tvar vyhledatelný, nezbývá než jej vytvořit. Výběr vhodného finálního tvaru je možné řešit použitím systému pravidel, který na základě značek výskytů kandidáta určí značku pro výsledný tvar. Každé pravidlo se skládá z podmínky, která se porovnává se seznamem výskytů, a z předpisu pro tvorbu výsledné značky. Při porovnávání v podmínce pravidla je možné vycházet ze značek společných pro všechny výskyty slova, z nejčastější značky, případně oba přístupy kombinovat. Ve většině případů si však vystačíme se společnou značkou. Kombinace přístupů je daleko využitelnější při interpretaci předpisu, kdy je užitečné umožnit nakopírování částí obou značek na daná místa. Například pro určení mluvnického čísla či stupně je vhodnější zkopírovat nejčastější značku, kdežto pokud budeme chtít použít stejný pád, jako mají všechny výskyty slova, zkopírujeme společnou.

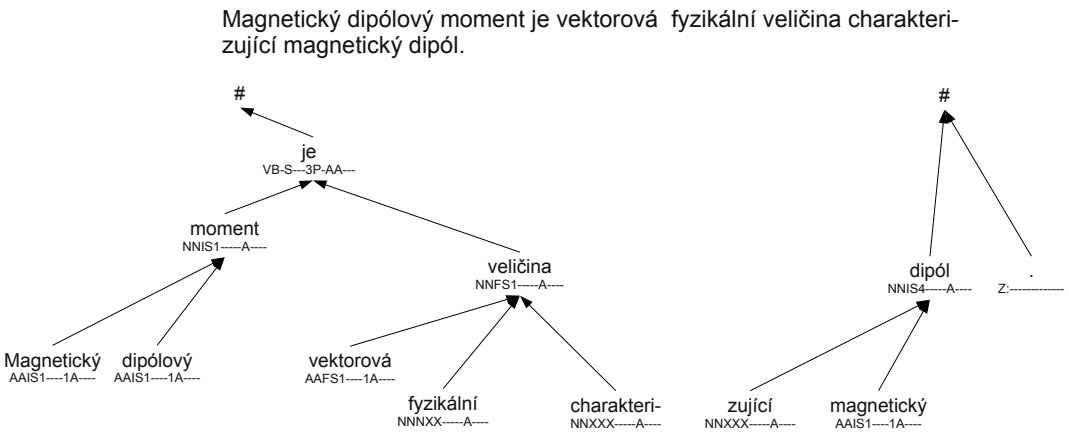
Ve chvíli, kdy máme určenou značku pro výsledné klíčové slovo, musíme mít nástroj, který požadovaný tvar slova vytvoří. Pro český jazyk je možné použít sady skriptů *Czech „Free“ Morphology* [9], která byla vytvořena v rámci PDT 1.0. Kromě morfologické analýzy nabízí tyto skripty také možnost z lemmatu a značky vytvářet žádané tvary slov. Vzhledem k tomu, že tento nástroj pochází se stejného projektu jako nástroje použité pro analýzu českých textů, používá stejných značek. Toho můžeme s využít při implementaci rozhraní pro úpravu slov.

4.6 Předúprava vstupních textů

Dřívější pokusy s automatickou extrakcí rejstříků ukázaly, že by bylo vhodné prozkoumat možnosti předúpravy vstupních textů ještě před tím, než projdou lingvistickou analýzou. Na výskyt některých znaků v textu nejsou totiž analyzátory připraveny, což může vyústit jednak ve špatně přiřazenou značku a jednak ve špatně vytvořený závislostní strom.

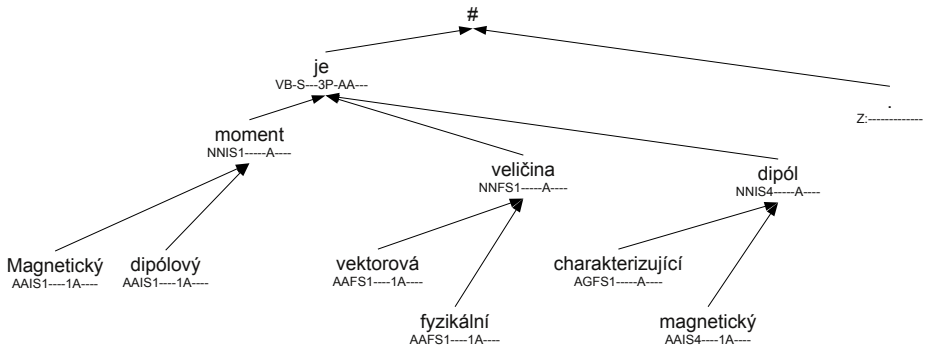
Jedním z takových jevů je rozdělování slov na koncích řádků spojovníkem[5]. Nástroje pro analýzu obvykle rozdělené slovo vyhodnotí jako dvě slova. Vzhledem k tomu, že tato slova potom nejsou nalezena ve slovnících, je jim přiřazen neznámý slovní druh, případně jsou označena jako podstatná jména. Pro ukázku uveďme dvě věty a pozorujme, jak si s nimi nástroje poradí. Analyzátoru PDT byl předložen úkol vytvořit závislostní strom věty „*Magnetický dipólový moment je vektorová fyzikální veličina charakterizující magnetický dipól.*“, ve které je slovo „*charakterizující*“ rozděleno na dva řádky. Výsledek je vidět na obrázku 4.5. Spojovník ve slově způsobil, že analýza našla dvě věty místo jedné. Obě části rozděleného slova určila jako podstatná jména. Stejnou značku navíc nesprávně přidělila slovu „*fyzikální*“. Pro srovnání je na obrázku 4.6 uveden strom automaticky vytvořený pro

[5]Spojovací čárka neboli spojovník je grafický znak v podobě vodorovné čárky kladené bez mezer mezi slova nebo jejich části. Užívá se, chceme-li vyjádřit, že jím spojené výrazy tvoří těsný (slovní nebo souslovný) celek. Spojovník se graficky i funkčně odlišuje od pomlčky. [11]



Obrázek 4.5: Ukázka závislostního stromu vytvořeného chybně kvůli rozdělenému slovu ve větě (PDT 2.0).

stejnou větu bez rozdělení slova. Obdobná věta byla značkována TreeTaggerem. Rozděleno

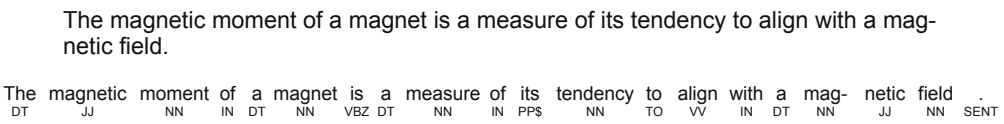


Obrázek 4.6: Správně vytvořený strom pro větu z obrázku 4.5 (PDT 2.0).

bylo tentokrát slovo „*magnetic*“. Vzhledem k tomu, že TreeTagger určuje pouze značky, nebyly napáchány tak velké škody. Přesto však byla identifikována dvě slova „*mag-*“ a „*netic*“, z čehož první bylo označeno jako podstatné a druhé jako přídavné jméno. Toto rozdělení jednoznačně povede k vyhledání nesmyslných kandidátů, jako jsou například „*mag-*“, „*netic mag-*“, „*netic field*“ a podobně[6].

Další nepříjemností jsou rovnice, které se mohou v dokumentu nacházet, ať už se jedná o rovnice zapsané přímo do čistého textu, nebo poškozené shluky znaků vzniklé z původních rovnic při OCR[7] převodu. Čísla v rovnicích jsou většinou určena správně jako číslovky. Stejně tak znaky pro matematické operace nedělají značkovačům problém. Kde ale narazíme, jsou názvy proměnných, které jsou typicky tvořeny krátkými, jedno až třípísmennými shluky. Ty jsou většinou označovány jako podstatná jména, předložky nebo číslovky v závislosti na tom, jestli obsahují číselný index, nebo ne. Je velice pravděpodobné, že proměnné

[6] Zde pochopitelně závisí na lingvistickém filtru (viz část 4.2).
[7] Optical character recognition



Obrázek 4.7: Ukázka věty označkované špatně v důsledku rozděleného slova ve větě (TreeTagger).

budou vybrány jako kandidáti na klíčová slova. Navíc je možné, že je hodnotící metody umístí na horních příčkách, neboť jejich referenční četnosti budou nízké.

Obdobné obtíže způsobují nadpisy, které jsou zapsány proloženě[8]. Jakkoli můžeme tento způsob zvýraznění textu považovat za nevhodný, v reálných textech jej autoři používají a je nutné se s ním vypořádat. Značkovače v takovém případě nemají možnost poznat, že se jedná o jedno slovo. Každé z písmen je obvykle označeno jako podstatné jméno. Je-li nad takto označenou větou vytvářen závislostní strom, bude pochopitelně nesmyslný.

Z předchozích odstavců plyne, že by bylo vhodné vytvořit filtrační nástroj, který by zpracoval text ještě před jeho analýzou a upravil, případně odstranil jeho nevhodné části. Takový filtr byl v rámci tohoto diplomového projektu vytvořen. Podrobnější informace o jeho implementaci a přínosu pro vyhledávání klíčových slov jsou uvedeny v kapitolách 5.6 a 6.5.

[8] Mezi každým písmenem textu je mezera.

Kapitola 5

Implementace

Systém byl implementován v souladu s myšlenkami nastíněnými v předchozí kapitole o návrhu. Po úvahách byl zvolen implementační jazyk Java. Ten zajistí snadný přenos výsledného řešení na různé platformy. Celý projekt navíc těží z objektového návrhu. Při volbě jazyka byla důležitá také dostupnost kvalitních nástrojů pro vývoj, což je v případě jazyka Java splněno.

V následujícím textu postupně zmíním detaily o realizaci jednotlivých částí systému a úskalí, která se během implementace objevila. Ta byla spojena především s použitím plně nepřímých kandidátů.

5.1 Lingvistická analýza

Lingvistická analýza bude svěřena externím nástrojům, které, jak bylo ukázáno dříve, se různí jednak úrovní zpracování a jednak svým výstupem. Proto bylo potřeba vytvořit jednotné rozhraní pro práci s dokumenty. Toto rozhraní bylo výrazně inspirováno výstupem nástrojů PDT 2.0, které nabízí množství informací uložené dostatečně obecným způsobem.

Množinu dokumentů v systému reprezentuje rozhraní *Corpus*. To umožňuje přidávat dokumenty jeden po druhém a načítat všechny soubory ze zvoleného adresáře. Navíc lze nastavit výběr dokumentů podmínit regulárním výrazem, kterému musí názvy souborů podléhat. Dokumenty v systému reprezentuje rozhraní *Document*. Úkolem tříd, které implementují tohoto rozhraní je zpracovat soubor ve formátu, ve kterém ho vytvořil příslušný nástroj. Každý dokument se dělí na odstavce (*Paragraph*), které se skládají z vět (*Sentence*). Věty ukládají jednotlivá slova do závislostního stromu, kterým je možno procházet. Zároveň s tím poskytuje přístup ke slovům v jejich původním pořadí. Slova jsou reprezentována rozhraním *Word*, které uchovává informaci o jejich původním tvaru, lemmatu, značce a funkci ve větě. Navíc obsahuje odkazy na rodičovské slovo a všechna dceřiná slova v závislostním stromu. Kromě klasického lemma může slovo vrátit ještě negované lemma, pokud bylo původní slovo v záporu. Negované lemma se používá prakticky v celém systému, aby nedošlo k záměně dvou slov, které jsou sice z morfologického hlediska stejné, ale mají odlišný význam.

Systém značek byl kompletně převzat z PDT 2.0, kde se používá pozičních značek o délce 15 znaků, přičemž předposlední dvě pozice jsou vyhrazeny pro budoucí použití [34]. To znamená, že mluvnické kategorie se ukládají do řetězce a každé kategorii je vyhrazena jedna pozice. Pokud není možné pro dané slovo gramatický jev určit, je na pozici vložen znak '-'. Podrobný popis značky je uveden v tabulce 5.1.

pozice	mluvnická kategorie
1	slovní druh
2	slovní poddruh, detail druhu
3	rod
4	číslo
5	pád
6	rod vlastníka
7	číslo vlastníka
8	osoba
9	čas
10	stupeň
11	negace
12	slovesný rod
13	rezervovaná pozice 1
14	rezervovaná pozice 2
15	varianta, styl

Tabulka 5.1: Mapování mluvnických jevů na pozice ve značce.

Jelikož je náročné pamatovat si pozice ve značce, byla pro zjednodušení vytvořena třída *Tag*, která značku zapouzdřuje a dává možnost přistupovat k jednotlivým gramatickým kategoriím přes funkce.

Rozhraní bylo implementováno pro všechny tři lingvistické analyzátory, které byly představeny v kapitole 4.

5.1.1 Pražský závislostní korpus

Jelikož rozhraní pro práci s dokumenty bylo do značné míry inspirováno výstupem nástrojů PDT 2.0, nevyskytly se při implementaci žádné větší obtíže. Byly vytvořeny třídy pro práci s dokumenty analytické vrstvy PDT, které obsahují vytvořené závislostní stromy a jednoznačné značky slov. Jedinou komplikací bylo kódování. PDT vyžaduje na vstupu soubory kódované znakovou sadou ISO/IEC 8859-2. Stejnou sadou jsou kódovány i výstupní soubory. Celý extraktor ovšem používá kódování UTF-8. Po několika pokusech s převodem kódování přímo při čtení souboru, které se ukázalo jako neúnosně pomalé, jsem se rozhodl přenechat konverzi externímu nástroji ještě před spuštěním extraktoru.

5.1.2 TreeTagger

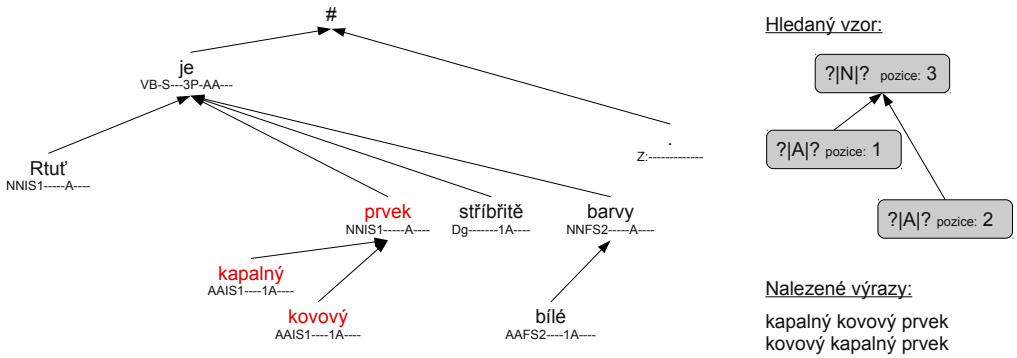
TreeTagger je pouze značkovač, který určuje lemmata a slovní druhy. Z toho důvodu byly závislostní stromy, které vyžaduje rozhraní pro reprezentaci věty, degradovány na pouhé sekvence. Tato úprava však funkci zbytku systému v zásadě nijak neohrožuje. Značky využívané TreeTaggerem jsou podstatně jednodušší než v PDT. Jedná se prakticky jenom o výčet slovních druhů. Proto bylo vytvořeno mapování mezi značkami TreeTaggeru na značky, které budou využívány interně v systému. Převodní tabulka se snaží zachovávat veškeré dostupné informace. Soubory označené TreeTaggerem rovněž neobsahují žádnou informaci o odstavcích. Obsahují však značky pro prázdné řádky. Dokumenty jsou proto děleny na odstavce v místech, kde se vyskytují dva a více prázdných řádků.

5.1.3 MiniPar

Také pro MiniPar jsou značky převáděny na vnitřní formát. Převodní tabulka je však mnohem jednodušší a prakticky obsahuje pouze značky pro slovní druhy. Členění do odstavců je řešeno obdobným způsobem jako u TreeTaggeru. MiniPar očekává ve vstupním souboru každou větu na samostatném řádku. Oddělovač mezi odstavci v označkovaných dokumentech tvoří prázdný strom vzniklý zpracováním volného řádku. Přestože MiniPar poskytuje oproti TreeTaggeru závislostní stromy vět, nebylo rozhraní pro práci s jeho dokumenty v systému dále využito. K tomuto rozhodnutí vedla dvě zjištění. Zaprvé MiniPar nedosahoval uspokojivé přesnosti při určování slovních druhů. Zadruhé lemmata jím vytvořená často rozšiřuje o slova z fráze, ve které se slovo nalézá. To výrazně ztěžuje další operace se slovem, jako je například vyhledání jeho četnosti v referenčním korpusu.

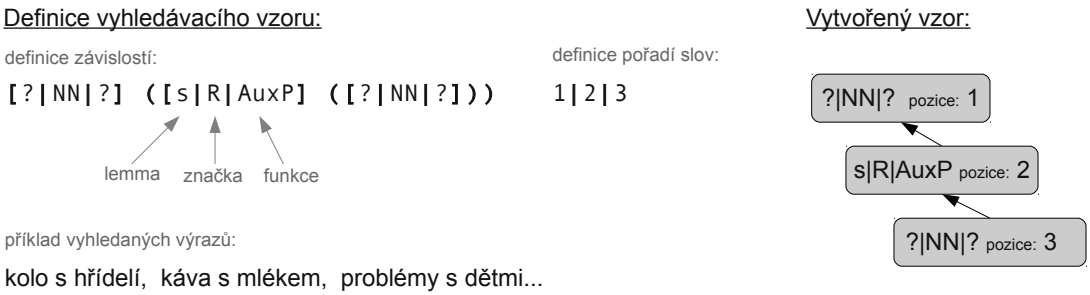
5.2 Výběr kandidátních klíčových slov

Výběr kandidátů na klíčová slova byl realizován vyhledáváním definovaných vzorů ve větných stromech. Vyhledávají se podstromy, které odpovídají nadefinované struktuře a jejichž uzly mají vlastnosti, které nejsou v rozporu s požadavky určenými ve vzoru. Ty je možné volně specifikovat v rozsahu od konkrétního slova v konkrétním tvaru až po libovolný uzel. Vzhledem k tomu, že poduzly ve stromu mohou mít různé pořadí, je třeba stanovit, jak budou slova ve výsledku uspořádána. I toto se určuje ve vyhledávacím vzoru. Každému slovu je přiřazena jeho pozice ve výsledném klíčovém slově. Některá slova je možno úplně vypustit, což dovoluje sloučit do kandidáta například dvě slova ze stejné úrovně zanoření ve stromu. Vyhledávání ve větném stromu ilustruje obrázek 5.1.



Obrázek 5.1: Ukázka vyhledávání definovaných vzorů ve větném stromu.

Vyhledávací vzory se definují v souboru v následujícím formátu. Každý vzor je vždy na zvláštním řádku a skládá se z části určující podmínky pro kořenové slovo vzoru a ze seznamu vzorů pro poduzly. U každého slova se určují pravidla pro jeho lemma, značku a funkci ve větě. Tato definice je uzavřena do hranatých závorek a za ní může následovat v kulatých závorkách seznam vzorů pro poduzly. Vzory v seznamu mají stejné složení a je možné je rekurzivně zanořovat. Pokud chceme některou z vlastností slova ignorovat, použije se místo ní znak '?'. V definičním souboru se nachází také seznam pozic, na kterých mají být slova umístěna. Pro objasnění ilustruje složení vyhledávacího vzoru a způsob jeho definice obrázek 5.2.



Obrázek 5.2: Ukázka definice vyhledávacího vzoru.

Před vyhledáváním je každý vzor předkompilován, aby byla práce s ním co možná nejefektivnější. O vyhledání kandidátů se v systému stará třída *CandidateExtractor*, která umožňuje vyhledat výskyty kandidátů jak v celém korpusu, tak v dokumentu, odstavci, či pouze ve větě. Výskyty jsou na závěr spojeny podle jejich lemmat a výstupem vyhledávání je mapa, ve které je ke každému lemmatu uložen kandidát, který zapouzdřuje seznam všech svých výskytů. O každému výskytu se eviduje odkaz na větu, odstavec, dokument a korpus ve kterém byl nalezen. Dále odkaz na vzor, na základě kterého byl vybrán, a pochopitelně také vlastní slova. Pokud by došlo k situaci, že dva vyhledávací vzory odpovídají stejnému výskytu, bude uložen jenom jeden z nich, aby nedocházelo ke zkreslení četností.

Přestože jak pro češtinu tak pro angličtinu už existují slovnědruhové filtry použité v jiných pracích pro obdobné úkoly jako je vyhledávání klíčových slov [28], rozhodl jsem se pro účely této práce navrhnout vlastní. Statistiky o slovních druzích a vazbách mezi nimi byly získávány z dokumentů trénovací množiny. Každé klíčové slovo bylo vyhledáváno ve větách dokumentu, ze kterého pocházelo. Při nálezu se vypočítal minimální podstrom věty, který pokrýval všechna slova z žádaného výrazu. O každém uzlu se zaznamenaly dostupné informace. Na základě statistik výskytu různých druhů podstromů byly vytvořeny vyhledávací vzory. Vybrány byly ty vzory, které se ve výsledcích objevily významně vícekrát než ostatní. Vzory byly následně mírně ručně upraveny, aby vytvářely jednoznačnější pořadí slov. To výrazně napomůže úpravě do výsledných tvarů. Při volbě pořadí slov byla přídavná jména umístěna na začátek výrazu. Podstatná jména byla ponechána v původním pořadí, pokud jedno z nich nebylo v druhém pádě. V takovém případě bylo toto podstatné jméno zařazeno až na konec. Toto řazení vychází z úvahy, že ve výrazech tvořených dvěma podstatnými jmény, jako je například *teplota tání*, je to ve druhém pádě slovo závislé, které svůj pád nemění.

Seznam posloupností slovních druhů, které filtry vyberou pro češtinu je uveden v tabulce 5.2, pro angličtinu potom v tabulce 5.3. Mnou vytvořený filtr má, jak se dalo očekávat, s těmi existujícími mnoho společného. Nicméně přínos metody, kterou jsem použil, je v přímém vytvoření vzorů zohledňující závislosti ve větách. Další výhodou mého postupu je, že vzory jsou vybírány s přihlédnutím k chybám, kterých se může dopustit analyzátor. Tato skutečnost je patrná mezi anglickými vzory, kde bylo vybráno například podstatné jméno následované slovesem. Toto sloveso je však většinou jen špatně rozpoznané podstatné, nebo přídavné jméno.

Při pokusech s vyhledáváním kandidátů se ukázalo, že je vhodné implementovat také

vzor	příklad klíčového slova
N	kryptografie
NN	teplota tání
NNN	index absorpce energie
NRN	odolnost proti přehýbání
AN	pyrotechnický zpožďovač
ANN	koncentrační součinitel vulkanizace
NAN	polovina vzorkovací frekvence
AAN	měrný povrchový odpor
DAN	oxidačně degradovatelný plast
V	laminovat

Tabulka 5.2: Seznam posloupností slovních druhů, které jsou vybírány českými pravidly pro hledání kandidátů (A - přídavné jméno, D - příslovce, N - podstatné jméno, R - předložka, V - sloveso).

vzor	příklad klíčového slova
N	endoscope
NN	emergency window
NNN	surface erosion control
NRN	source of illumination
AN	atmospheric pressure
ANN	thermal insulation material
NAN	wood preservative paste
AAN	remote visual testing
VN	tear strength
NV	clay slip
V	testing

Tabulka 5.3: Seznam posloupností slovních druhů, které jsou vybírány anglickými pravidly pro hledání kandidátů (A - přídavné jméno, N - podstatné jméno, R - předložka, V - sloveso).

seznam stopslov[1], který dokáže odfiltrovat z výsledků vyhledávání nevhodné výrazy. Tím se zabrání situacím, kdy lingvistický analyzátor selže a určí například předložku nesprávně jako podstatné jméno. Implementován byl klasický stop seznam a stop seznam s regulárními výrazy, který dovoluje filtrovat slova i podle délky. To je užitečné pro odstranění proměnných ze vzorců, které jsou zpravidla označeny jedním až dvěma písmeny.

5.3 Sjednocení klíčových slov

Pro sjednocování kandidátů podle synonym byla navržena třída *CandidateUnifier*, které je třeba nastavit slovník synonym, s nímž má pracovat. Na vstup dostává mapu kandidátů. Výstupem výpočtů je tatáž mapa, jejíž prvky mají ovšem upravené odkazy na výskyty a která je případně obohacena o úplně nové kandidáty. To napovídá, že se jedná o část

[1]Stopslovo je slovo, které se v daném jazyce vyskytuje často, ale nenese žádnou významovou informaci, má zpravidla pouze syntaktický význam. Typicky se jedná o spojky, předložky atp. [3]

systému, kterou je možné vypustit, aniž by přestal být funkční.

Ke sjednocení kandidátů je možné použít jakéhokoliv slovníku synonym, pro který byla vytvořena obálka implementující rozhraní *Thesaurus*. Toto rozhraní obsahuje funkci pro zjištění seznamu synonym pro dané slovo, u kterého je možné navíc specifikovat slovní druh, čímž může být navrácený seznam zpřesněn, pokud slovník tuto možnost podporuje. Byly vytvořeny třídy obalující český a anglický thesaurus z projektu OpenOffice.org a anglický slovník WordNet. Oba anglické thesaury umožňují vyhledání zpřesnit slovním druhem.

Přítomnost této části v systému si vyžádala úpravy ve způsobu ukládání kandidátů. Jelikož se ke každému kandidátu ukládá seznam jeho výskytů v textu, je při jejich sjednocování namísto pouhého zvětšení četností tento seznam rozšířen o výskyty, které jsou synonymicky ekvivalentní. V seznamu je potom třeba rozlišovat, jestli se jedná o přímý výskyt kandidáta, nebo o nepřímý výskyt vzniklý kopírováním. Lemmata nepřímých výskytů se totiž od původního lemma kandidáta liší. Proto nelze nepřímé výskyty použít například pro vyhledání správného tvaru slova přímo v textu (viz kapitola 4.5). Z tohoto důvodu byly zavedeny dva seznamy, každý pro jeden typ výskytů. Stejně tak četnosti, které vycházejí z délek těchto seznamů, je možné počítat jako přímé, nepřímé a souhrnné.

Z optimalizačních důvodů byl algoritmus navržený pro sjednocování kandidátů mírně upraven. Na začátku se vytvoří synonymické obměny pro každého kandidáta a výsledek se uloží do struktury, která mapuje obměněné varianty na seznam původních lemmat. Následně pouhým průchodem této mapy jednoduše zjistíme, které obměny jsou společné pro více kandidátů. Jestliže se obměna mezi kandidáty již vyskytuje, dojde pouze k nakopírování nepřímých výskytů. V opačném případě je, za předpokladu že je povolena tvorba plně nepřímých kandidátů, nejprve vložen nový kandidát. Tímto postupem se postupně sjednotí všechny výskyty.

5.4 Výběr klíčových slov

I při realizaci výběru výsledné množiny klíčových slov byl kladen důraz na univerzálnost řešení. Bylo vytvořeno rozhraní *Algorithm*, které je určeno pro implementaci jednotlivými algoritmy. Toto rozhraní obsahuje jedinou funkci *execute*, které je předána mapa kandidátů. Právě v těle této funkce musí proběhnout hodnocení kandidátů. Jako výstup funkce je očekávána opět mapa. Ta už ale propojuje pouze lemmata s jejich ohodnocením. Rozhraní *Algorithm* bylo implementováno pro 9 algoritmů, zmíněných v kapitole 2.

Celý proces výběru klíčových slov řídí třída *TermEvaluator*. V této třídě je potřeba zaregistrovat všechny algoritmy, které plánujeme při výběru použít. Při registraci se každému algoritmu přidělí jeho jméno, podle kterého lze později vyhledat jím vypočítané výsledky. Zároveň se v této třídě nastavuje strategie pro kombinování výsledků. Strategiím je určeno rozhraní *RankingCombinationMethod*. Podobně jako rozhraní pro algoritmy předepisuje jen jednu metodu. Ta má dokonce i stejný výstup. Pouze na vstupu jsou jí předány mapy s výsledky vypočtenými všemi algoritmy. V rámci této práce byla implementována strategie volení, která je popsána v části 2.3.

Hodnotící metody obvykle potřebují pro svou práci více než jenom znalost souhrnné frekvence výskytů slov ve zpracovávaném korpusu. Mnohdy se používají četnosti počítané zvlášť pro dokumenty, případně odstavce, které jsou nějakým způsobem porovnávány s celkovým počtem slov. Počet výskytů v dané textové jednotce můžeme poměrně jednoduše získat ze seznamu uloženém v kandidátovi. S celkovým počtem slov je situace složitější. Tuto informaci o sobě sice rozhraní pro všechny textové jednotky poskytují, nicméně hodnotu je nutné při každém dotazu na ni znova přepočítat. Rozhraní totiž umožňují přidávat

obsah dynamicky. S ohledem na tento fakt bylo vytvořeno rozhraní *WordCountStats*, sloužící pro předpočítání údajů o počtech slov. Třídy implementující toto rozhraní poskytují rychlý přístup k statistikám z korpusu a dokumentu.

Pro algoritmy, které používají údaje o frekvenci slov z referenčního korpusu, byly vytvořeny soubory s četnostmi unigramů a bigramů. Jak uzkazuje tabulka 5.4, velikosti původních souborů byly příliš velké na to, aby mohly být použity. Obsahovaly mnoho nesmyslných shluků znaků, které se objevily v korpusech a při automatické tvorbě četností byly identifikovány jako slova. Proto byly soubory filtrovány. Nejdříve byla odstraněna všechna slova, která obsahovala více než jednu pomlčku uprostřed slova nebo jiné znaky než písmena. Potom byl soubor četností označkován nástrojem pro příslušný jazyk a vybrány byly jen ty kombinace slovních druhů, které se mohly vyskytnout v množině kandidátů. Pro ty se namísto původních tvarů ukládala lemmata. Toto filtrování velikost četností značně zmenšilo. Přesto však byly vytvořeny ještě soubory, ve kterých byla každá četnost snížena o 3 a slova s hodnotou menší než 1 byla vyřazena. Takto upravené soubory byly použity pro finální verzi extraktoru.

soubor četností	původní velikost	velikost po filtrování	velikost po filtrování s minimální četností 3
české unigramy	68.7 MB	20.8 MB	6.2 MB
české bigramy	1.8 GB	141.3 MB	52.3 MB
anglické unigramy	98.4 MB	33.6 MB	11.7 MB
anglické bigramy	1.4 GB	67.3 MB	54.0 MB

Tabulka 5.4: Přehled velikostí souborů s četnostmi slov po úpravách.

Použití unigramů a bigramů bylo zvoleno jako vhodný kompromis mezi poskytnutím přesné informace a množstvím použité paměti při běhu extraktoru. V systému jsou četnosti reprezentovány rozhraním *Frequencies*, které bylo implementováno pro každý druh četností zvlášť. Hodnoty pro výrazy delší než dvě slova jsou odhadovány stejným způsobem, jako byl opsán v části 4.4. Tento přístup zapouzdřuje třída *MultigramRefFreq*, která umožňuje spojit více četností pro různé délky n-tic. Při dotazu na frekvenci výrazu potom zkouší najít hodnotu v souboru s nejdelšími možnými n-ticemi. Pokud není nalezena, vyhledává postupně podčásti výrazu v četnostech pro kratší n-tice.

Určování referenčních četností komplikují nepřímé výskyty kandidátů, protože všechny nemusí obsahovat stejná lemmata. U kandidátů s alespoň jedním přímým výskytem se jednoduše použije jeho původní lemma. Pro plně nepřímé kandidáty je četnost vypočítána jako průměr z hodnot určených pro lemmata jednotlivých výskytů.

Při volbě způsobu uložení do paměti bylo dbáno především na rychlost načtení. Z hlediska času je pro přidávání prvků nejrychlejší strukturou v Javě *ArrayList*. Proto byl obsah souborů s četnostmi abecedně seřazen a načítán do pseudomapy vytvořené nad *ArrayListem*. Vzhledem k abecednímu uspořádání prvků pak bylo možné vyhledávat v seznamu binárně. Úspora paměti je u tohoto řešení bohužel nulová. Doba načítání je však extrémně krátká. Pro paměťově šetrné řešení by bylo nutné implementovat některou z metod uvedených v návrhu.

Při práci s jednotlivými hodnotícími algoritmy vyvstala omezení na jejich použití. Některé z nich totiž pro správnou funkci vyžadují širší kontext a není možné je použít na vyhledání klíčových slov v jednotlivém dokumentu. V praxi to může být problém, pokud z nějakého důvodu nemůžeme zpracovat celý balík dokumentů naráz. Řešením, které však

v této práci nebylo implementováno, je uchovávat důležité informace o dříve zpracovaných dokumentech. Ucelený přehled prostředků, které jednotlivé algoritmy vyžadují, je uveden v tabulce 5.5.

algoritmus	pracuje jen nad více dokumenty	používá referenční četnosti
Term Frequency	ne	ne
TF IDF	ano	ne
Residual IDF	ano	ne
Domain Consensus	ano	ne
Weirdness	ne	ano
Likelihood ratio	ne	ano
BM25	ano	ne
Lexical Cohesion	ne	ne
C Value	ne	ne

Tabulka 5.5: Přehled algoritmů a jejich nároků.

Výstupem této části systému je seznam finálních klíčových slov, které reprezentuje třída *Term*. Ta uchovává lemma klíčového slova, upravený tvar, výsledky hodnocení každého algoritmu a souhrnný výsledek vzniklý kombinací. Tyto informace jsou pro další zpracování dostačující. Samotná třída *TermEvaluator* dovoluje omezit výběr klíčových slov podle tří kritérií. Lze vybrat buď určité procento nejlépe hodnocených, pevný počet nejlépe hodnocených, nebo klíčová slova, která mají hodnocení v rozsahu určitých hodnot. V posledním případě je možné nechat horní hranici rozsahu otevřenou. Všechny tyto druhy výběru je možné omezit jen na klíčová slova z určitého dokumentu. Případné sofistikovanější způsoby výběru by bylo možné specifikovat mimo tuto třídu. Posloužit k tomu mohou již vytvořené pomocné metody, které dovolují řadit výstup podle výsledků jednotlivých algoritmů.

5.5 Úprava slov do výsledných tvarů

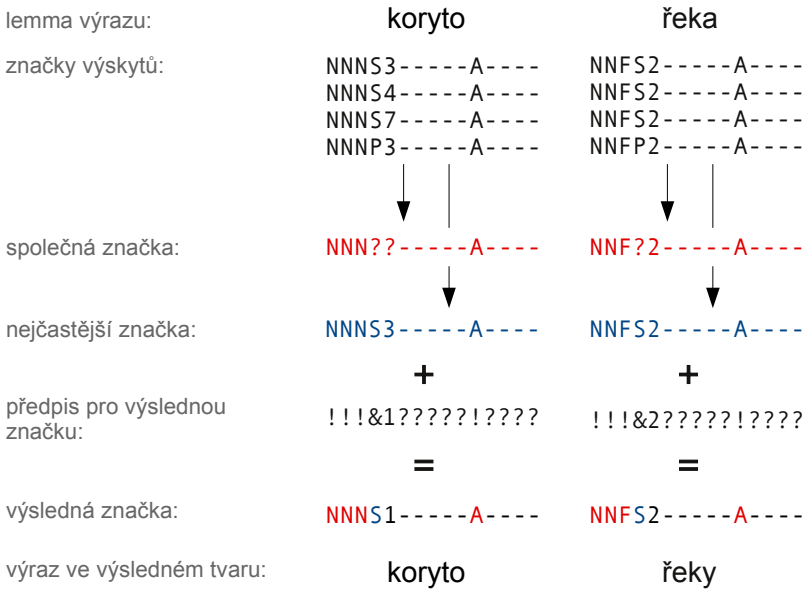
Jak už bylo naznačeno výše, úprava klíčových slov je nutná zejména v českém jazyce, kde jsou u víceslovných výrazů základní tvary zcela nevhodné. Pro anglický jazyk není problém tolik palčivý. Z těchto důvodů byla realizována úprava pouze pro české texty. Pro anglická klíčová slova je určování výsledného tvaru značně zjednodušené.

Implementace proběhla zcela v souladu s návrhem z části 4.5 předchozí kapitoly. Probíhá ve dvou fázích. První fází je pokus o vyhledání výskytů v prvním pádě v původním textu. V druhé fázi se výsledné tvary vytváří pro ta klíčová slova, pro která bylo hledání neúspěšné. Značka tvaru, který má být vytvořen je určena systémem pravidel. Každé pravidlo se skládá z podmínky pro použití a předpisu pro vytvoření tvaru. Podmínka pravidla má syntaxi podobnou vyhledávacím vzorům zmíněným v části 5.2. Omezuje použití pravidla specifikováním lemma, značky a funkce každého slova. Při vyhodnocování podmínky se značka porovnává se značkou společnou pro všechny výskyty. Protože v této fázi extrakce už mají slova pevné uspořádání, nelze v podmínce definovat závislostní vztahy slov.

Předpis pro tvorbu výsledné značky může obsahovat přesně určené hodnoty, kopie hodnot ze značky společné pro všechny výskyty a kopie hodnot z nejčastější značky. Postup tvorby takových značek a interpretaci pravidla znázorňuje obrázek 5.3. Samozřejmě je také možné hodnotu nespecifikovat. Sémantiku znaků, které je možno v předpisu použít shrnuje tabulka 5.6.

znak	význam
?	libovolná hodnota
!	hodnota společná pro všechny výskyty kl. slova
&	nejčastější hodnota ve výskytech kl. slova
jiný znak	interpretován přímo jako určení hodnoty

Tabulka 5.6: Sémantika znaků v předpise pro určení značky pro úpravu slova.



Obrázek 5.3: Ukázka interpretace pravidla pro určení výsledného tvaru.

Po výběru a interpretaci předpisu je vytvořená značka společně s lemmatem klíčového slova předána nástroji pro tvorbu tvarů slov. Ke komunikaci s takovým nástrojem bylo navrženo rozhraní *Inflector*. Jeho jedinou implementací je *FMorphInflector* zapouzdřující sadu skriptů *Czech „Free“ Morphology* [9]. Pokud se nepodaří vytvořit pro nějaké slovo požadovaný tvar, ponechá se lemma.

Při tvorbě pravidel pro úpravu bylo dodrženo několik zásad, které vedou ke správně určenému výslednému tvaru. U každého slova pochopitelně musel být zachován slovní druh a rod. Dál se zachovává negace, aby nedocházelo ke změně významu. Číslo se určuje podle nejčastějšího čísla ve výskytech, což zaručuje vysokou pravděpodobnost správného určení. Přídavným jménům se kopíruje rod podstatných jmen, na kterých závisí. Všem slovům až na dvě výjimky je přiřazen první pád. První z výjimek jsou dvě podstatná jména spojená předložkou, kdy je druhému podstatnému jménu vybrán nejčastěji se objevující pád. To vychází z předpokladu, že jeho pád určuje předložka, se kterou se pojí, a nebude se měnit. Druhou výjimkou jsou dvě podstatná jména stojící vedle sebe. Pravidla pro výběr kandidátů zajistí, že v takovém případě je řídicí podstatné jméno řazeno jako první. Proto je vždy druhému, závislému, slovu nastavován druhý pád. Ten je totiž nejčastějším pádem, který pojí dvě podstatná slova do jednoho výrazu.

Jakákoliv úprava je bohužel nemožná pro klíčová slova vzniklá z plně nepřímých kandi-

dátů. Vyhledat tvar v textu je nemožné, protože neexistují přímé výskyty. Úpravu pomocí systému pravidel zase znemožňuje neznalost značek jednotlivých slov. To je sice možné řešit opětovnou analýzou, nicméně by tím došlo k výraznému zesložitění. Po přihlédnutí k experimentálnímu charakteru nepřímých kandidátů jsem se rozhodl toto řešení neimplementovat a ponechat plně taková klíčová slova bez úpravy.

Pro anglické výrazy nebyl systém pravidel vytvořen. Úprava probíhá jen ve značně zjednodušené míře. Pokud se v textu vyskytuje klíčové slovo jen v jednom jediném tvaru, je tento tvar určen jako výsledný. Tím je například alespoň částečně dosaženo výběru vhodného čísla, pokud je v něm termín zaužíván.

5.6 Předúprava vstupních textů

Předúprava vstupních výrazů byla implementována jako samostatný program, který na zbytku systému nijak nezávisí. Může proto být jednoduše využit i pro jiné projekty. Klíčovým požadavkem byla modulárnost, aby nebyl problém program do budoucna upravovat a rozšiřovat. Celá úprava textů byla rozdělena do několika nezávislých částí, modulů, které lze spouštět postupně za sebou. Kdyby byly moduly vytvořeny jako klasické třídy, nezačal by jeden práci dřív, než všechny předchozí moduly skončí. Navíc by to znamenalo celý zpracovávaný soubor udržovat v paměti. Řešením těchto nepříjemností je umístit každý modul do vlastního vlákna.

Program pro předúpravu byl tedy vytvořen jako vícevláknový, přičemž jednotlivá vlákna spolu komunikují pomocí pipeline. O celý průběh úpravy textu se stará třída *Cleaner*. U ní se registrují moduly zpracovávající text. Třída moduly řadí postupně do řetězce, který začíná startovacím modulem, jehož úkolem je pouze číst vstupní soubor a posílat jeho obsah dál. Na konci řetězce stojí terminální modul, který naopak jenom zapisuje upravený text do výstupního souboru. Toto řešení vykazuje velmi dobrou rychlost zpracování, protože vlákna pracují paralelně.

Kromě startovacího a terminálního moduly byly implementovány následující moduly pro filtraci textu:

- *odstranění vzorců* - odstraní ty řádky, jejichž obsah identifikuje jako vzorec. Tento modul je schopen odstranit pouze ty vzorce, které stojí na řádku samostatně (nejsou obklopeny textem). Vzorce rozeznává pomocí vypočítaného poměru písmen a bílých znaků k délce řádku. Podmínky pro označení řádku jako vzorce jsou následující[2]:

```
if (letterRatio <= 0.5) {
    return true;
} else if (letterRatio < 1.0) {
    if (spaceRatio == 0)
        return true;
    if (spaceRatio >= 0.25)
        return true;
}
return false;
```

  I přes svoji jednoduchost je tento způsob výběru vzorců velmi účinný.

[2] *letterRatio* je poměr písmen k délce řádku a *spaceRatio* poměr bílých znaků k délce řádku

- *odstranění spojovníků* - je-li na konci řádku, který tento modul přijme, spojovník, odstraní ho. Zároveň s ním odstraní i znak konce řádku, což způsobí slití rozděleného textu zpět. Ze začátku přijatého hned po odstranění spojovníku jsou odebrány bílé znaky, které by případně mohly stát v cestě slití textu.

- *odstranění čísel kapitol* - odstraňuje ze začátku řádků sekvence číslic a teček až po první bílý znak. Sekvence musí začínat číslem, nesmí obsahovat dvě tečky za sebou a maximální délka posloupnosti číslic nesmí být větší než N. Hodnota pro N byla experimentálně stanovena na 2. Tato sada pravidel odpovídá formátu vysledovaném u několika publikací.

Implementované moduly je třeba brát jako vzorek mnoha možných metod předúpravy textu, na kterém bude testován přínos celé myšlenky. V případě, že se tento postup osvědčí, může být program rozšířen o další moduly.

Kapitola 6

Výsledky experimentů

Všechny implementované části byly postupně testovány tak, aby mohl být určen přínos každé z nich. Podle dosažených výsledků byla zvolena výsledná konfigurace systému, na které byl proveden podrobnější test.

Při testování bylo ve většině případů použito porovnání lemmat klíčových slov. Tento test bude v této kapitole považován za implicitní a je použit vždy, kdy není uvedeno jinak. Pro přesnější porovnání výsledků bylo provedeno přesné porovnání výsledných tvarů a porovnání akceptující různé uspořádání slov. Test uvažující i synonyma nakonec kvůli jeho slabé vypovídací hodnotě a silným ovlivněním kvalitou použitého slovníku použit nebyl.

Výsledky každého z testů budou vyneseny do dvou grafů. Jednak do grafu přesnosti v závislosti na počtu vybraných klíčových slov a jednak do grafu úplnosti v závislosti na počtu vybraných klíčových slov. Všechny hodnoty v nich budou uvedeny v procentech.

Z dostupných norem bylo pro účely testování vybráno 53 dokumentů v českém jazyce a 40 dokumentů anglických. Množina článků z konferencí v anglickém jazyce byla za účelem zkrácení doby běhu testů omezena na 100 dokumentů. Počet klíčových slov, vůči kterým se výsledky budou porovnávat, je uveden v tabulce 6.1. Je patrné, že v článcích je jejich hustota velice nízká. Tím se stává jejich vyhledání velmi těžkým úkolem a dá se předpokládat, že dosahované přesnosti budou relativně nízké.

soubor dokumentů	celkový počet referenčních kl. slov	průměrný počet kl. slov
české normy	11 684	220
anglické normy	8 550	213
anglické články	395	5

Tabulka 6.1: Počty referenčních klíčových slov v souborech dokumentů pro testování.

6.1 Výběr kandidátních klíčových slov

Nejprve byly zkoumány vlastnosti široké množiny vybraných kandidátů. Tabulky 6.2 a 6.3 ukazují, jakou část referenčních klíčových slov který vzor v slovnědruhovém filtru pokrývá. Je vidět, že s drobnými odlišnostmi mají nejzásadnější podíl na pokrytí klíčových slov v každém testovacím souboru dat podstatná jména, jejich dvojice a přídavné jméno následované podstatným. Tyto tři vzory dohromady pokrývají zhruba 60 % správných klíčových slov. Všechny ostatní jsou pak zastoupeny výrazně menší měrou. Celkové pokrytí je rovněž pro všechny testovací soubory více méně shodných 80 %. To znamená, že 20 % klíčových slov při takovémto nastavení filtru nemůže být vůbec nalezeno. Při vyhodnocování shody bylo v těchto testech použito porovnávání lemmat.

vzor	české normy
N	25,53 %
NN	10,02 %
NNN	1,01 %
NRN	2,26 %
AN	31,38 %
ANN	2,45 %
NAN	2,12 %
AAN	3,63 %
DAN	0,37 %
V	0,24 %
celkem	79,04 %

Tabulka 6.2: Pokrytí referenčních klíčových slov vzory v českém slovnědruhovém filtru (A - přídavné jméno, D - příslovce, N - podstatné jméno, R - předložka, V - sloveso).

vzor	anglické normy	anglické články
N	26,89 %	21,95 %
NN	21,79 %	29,56 %
NNN	3,14 %	6,30 %
NRN	1,56 %	0,43 %
AN	13,54 %	12,17 %
ANN	2,96 %	3,47 %
NAN	0,90 %	0,86 %
AAN	0,37 %	0,86 %
VN	5,08 %	1,08 %
NV	2,21 %	2,39 %
V	3,93 %	0,43 %
celkem	82,44 %	79,56 %

Tabulka 6.3: Pokrytí referenčních klíčových slov vzory v anglickém slovnědruhovém filtru (A - přídavné jméno, N - podstatné jméno, R - předložka, V - sloveso).

Přestože 20 % klíčových slov nebude tímto filtrem nikdy odhaleno, nemá význam jej rozšiřovat. Vzory ostatních klíčových slov se natolik různí, že pro zaručení plného pokrytí referenční množiny bychom se dostali do situace, kdy by filtr ztrácel smysl. Celkové počty kandidátů vybraných na základě zvolených vyhledávacích vzorů jsou uvedeny v tabulce 6.4.

soubor dokumentů	celkový počet vybraných výrazů
české normy	143 816
anglické normy	29 377
anglické články	137 364

Tabulka 6.4: Celkové počty všech výrazů vybraných na základě vyhledávacích vzorů.

6.2 Výběr klíčových slov

Při hledání vhodných algoritmů pro ohodnocení kandidátů byly zkoumány všechny metody zmíněné v kapitole 2, kromě metody *C-Value*. Ta byla vynechána kvůli své vysoké náročnosti na výpočet, kterou celý systém brzdila. Pro přehledné porovnání všech metod na jednotlivých trénovacích korpusech bylo počítáno průměrné umístění referenčních klíčových slov řazených podle získaného ohodnocení v seznamu. Výsledek je zobrazen v tabulce 6.5, která obsahuje průměrné pozice normalizované do intervalu 0 až 1.

algoritmus	české normy	anglické normy	anglické články	suma	průměr
BM25	0.0119	0.0330	0.0812	0.1261	0.0420
LexicalCohesion	0.0059	0.0285	0.0952	0.1296	0.0432
TFIParF	0.0063	0.0285	0.0956	0.1304	0.0435
TFIDF	0.0063	0.0293	0.0954	0.1310	0.0437
TF	0.0064	0.0283	0.1034	0.1381	0.0460
Weirdness	0.0296	0.0628	0.2834	0.3758	0.1253
DomainConsensus	0.0117	0.0422	0.3425	0.3964	0.1321
ResidualIDF	0.0451	0.0709	0.4803	0.5963	0.1988
LikelihoodRatio	0.0469	0.0762	0.4834	0.6065	0.2022

Tabulka 6.5: Tabulka průměrného umístění referenčních klíčových slov pro jednotlivé algoritmy a korpusy. Uspořádáno podle výsledného průměru.

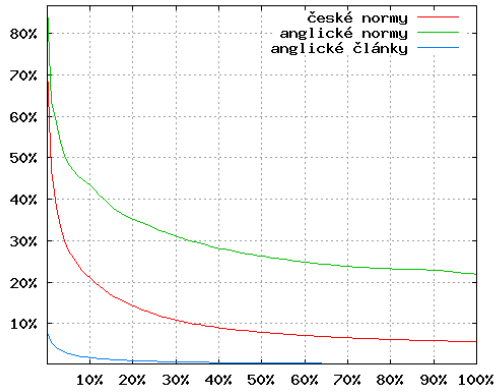
Z tabulky je vidět, že pro všechny trénovací dokumenty si velmi dobře vedlo prvních pět metod. Ty průměrně umístily všechna žádaná klíčová slova okolo prvních 5 % svého výstupu. Zbytek dosáhl znatelně horšího výsledku především na množině anglických článků. Pro výslednou konfiguraci systému a další testy byl nakonec výběr algoritmů zúžen na první čtyři, které už určitým způsobem ve výpočtech hodnotu *Term Frequency* zohledňují. Tyto algoritmy byly při dalším použití kombinovány metodou *Weighted Voting*. Přiřazené váhy byly určeny na základě maximálních přesností dosažených na trénovací sadě českých norem. Celkový poměr je však zhruba stejný i pro ostatní korpusy. Jednotlivé váhy jsou uvedeny v tabulce 6.6.

algoritmus	zvolená váha
BM25	0.33
Lexical Cohesion	0.47
TF - IPF	0.47
TF - IDF	0.48

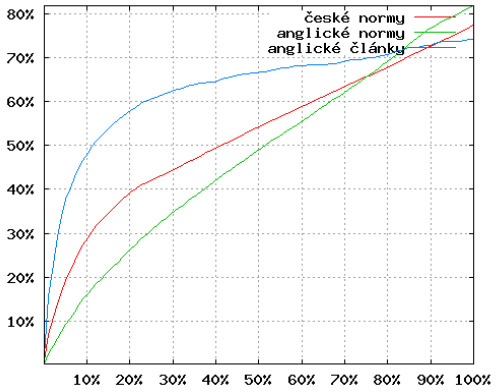
Tabulka 6.6: Váhy jednotlivých algoritmů pro kombinaci metodou *Weighted Voting*.

Pro zvolenou kombinaci algoritmů byl proveden test základního vyhledání klíčových slov. Výstupní grafy jsou na obrázcích 6.1 a 6.2. Z grafu přesnosti je možné odhadnout charakter jednotlivých souborů dat. Obě křivky pro normy mají podobný tvar, ovšem na anglických normách bylo dosaženo vyšší přesnosti. Na základě faktu, že tato přesnost se i při akceptování všech nabídnutých slov nedostane pod 20 %, lze usuzovat, že anglické normy obsahují celkově méně textu. Počet klíčových slov však zůstává stejný a proto je vyhledávání snazší. Opačným extrémem je křivka přesnosti pro anglické články. Ty obsahují,

jak už bylo řečeno dříve, velmi malý počet referenčních klíčových slov. Proto není ani pro výběr prvních nabídnutých výrazů přesnost nijak vysoká. Pohled na grafy úplnosti však prozrazuje, že v rámci možností je většina požadovaných klíčových slov umístěna v prvních 20 % nabídnutého seznamu.



Obrázek 6.1: Graf přesnosti v závislosti na množství vybraných klíčových slov pro kombinaci algoritmů.



Obrázek 6.2: Graf úplnosti v závislosti na množství vybraných klíčových slov pro kombinaci algoritmů.

Vzhledem k tomu, že grafy přesnosti strmě padají a může být problém vyčíst přesné hodnoty, uvádím pro doplnění ještě tabulku přesností 6.9. Obsahuje hodnoty pro prvních 100, 200 a 300 vybraných klíčových slov. Z tabulky je patrné, že vybraná kombinace algoritmů si vede ve vyhledávání v normách obstojně.

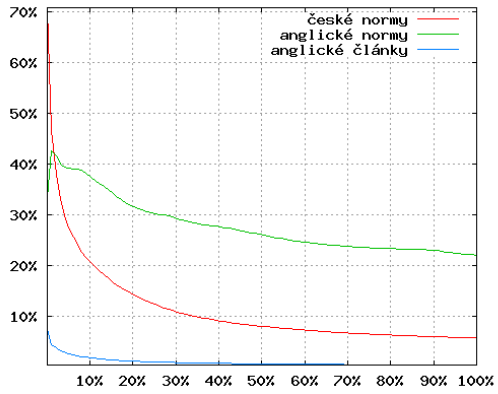
soubor dokumentů	100	200	300
české normy	72,0 %	69,0 %	65,0 %
anglické normy	77,0 %	69,0 %	63,0 %
anglické články	9,0 %	10,0 %	8,6 %

Tabulka 6.7: Přesnosti dosažené pro prvních 100, 200 a 300 vybraných klíčových slov.

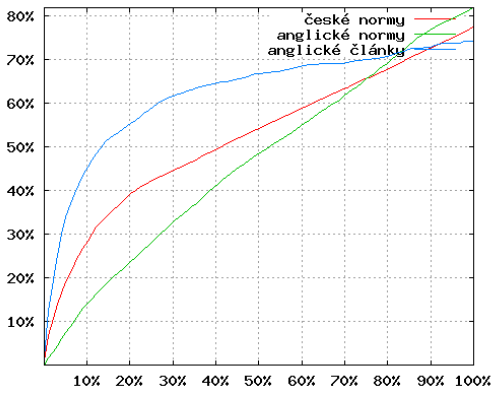
6.3 Sjednocení klíčových slov

Sjednocování klíčových slov na základě synonym nepřineslo dle výsledků testů žádný užitek. Testy byly rozděleny na test pouhého sjednocování a na test sjednocování s tvorbou plně nepřímých kandidátů. Výsledky obou však byly téměř totožné a proto uvedu grafy pouze pro první z nich.

Z obrázků 6.3 a 6.4 je vidět, že u českých norem nedošlo prakticky k žádné větší změně. Podrobné zkoumání vybraných klíčových slov a porovnávání s výstupem vytvořeným čistě pro kombinaci algoritmů přineslo zjištění, že žádný plně nepřímý kandidát se nedostal na čelní pozice výstupního seznamu. Některá sjednocená víceslovná klíčová slova se posunula o pár pozic výše, nicméně neudála se žádná dramatická změna. Určitou část viny nese i použitý slovník synonym, který je příliš obecný.



Obrázek 6.3: Graf přesnosti v závislosti na množství vybraných klíčových slov pro sjednocení kandidátů.



Obrázek 6.4: Graf úplnosti v závislosti na množství vybraných klíčových slov pro sjednocení kandidátů.

U anglických norem došlo k výraznému propadu křivky přesnosti. Mezi nejlépe ohodnocená klíčová slova se dostaly poměrně obecné výrazy. Valnou většinu z nich tvořila slovesa. Dál docházelo k případům, kdy byla správně vybraná víceslovná klíčová slova vytlačena z čelních pozic jednotlivými slovy, které obsahovala. To je patrné i z grafu 6.3, kde se ze začátku přesnost mírně zvedá, až se dostane na své maximum 44 %.

Na článcích sjednocování také příliš úspěšné nebylo. Z grafů plyne, že žádaná klíčová slova byla mírně odsunuta na zadnější pozice. Dopředu se opět dostala slova, která se sice týkají daného tématu, ale nemají dostatečnou vypovídací hodnotu, aby mohla být použita jako klíčová slova.

V souhrnu se sjednocování klíčových slov nedá považovat za přínosné. Ani na jedné množině testovacích dat nepřineslo zvýšení úspěšnosti vyhledání. Pro vyhledávání klíčových slov v odborných textech tedy musíme tuto metodu označit za nevhodnou.

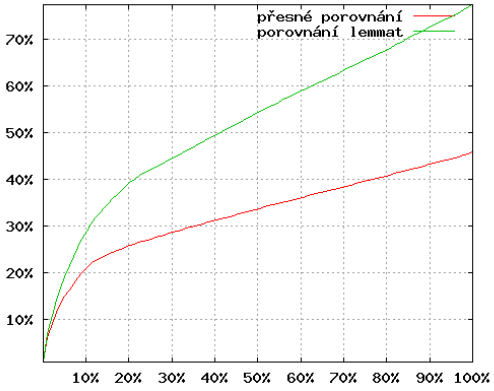
6.4 Úprava slov do výsledných tvarů

Vzhledem k odlišnostem úprav klíčových slov pro jednotlivé jazyky rozdělím i vyhodnocení provedených testů. Tato část se zabývá pouze zlepšením výběru vhodného tvaru a proto jsou hodnoty v testech porovnávajících lemmata zcela shodné s hodnotami základních testů z části 6.2.

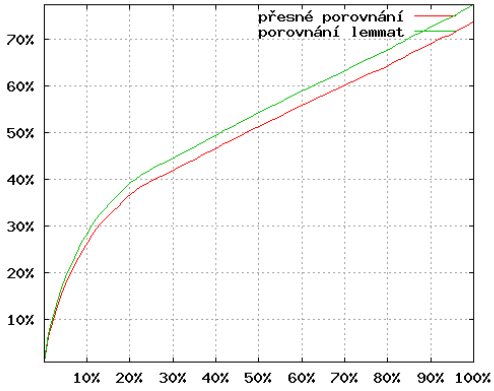
6.4.1 Česká klíčová slova

Pro všechna klíčová slova vyhledaná v českých normách se pro 34 % z nich podařilo nalézt základní tvar přímo v textu. Vytvořená pravidla pro úpravu pokryla 65 %. Pouze pro necelé 1 % klíčových slov se nepovedlo nalézt vhodné pravidlo. To byly ojedinělé případy, kdy analyzátor neurčil správně slovní druh slova, což způsobilo nesprávný výpočet společné značky. Kvůli tomu nemohlo být vybráno žádné z pravidel.

Pro měření úspěšnosti byl použit jednak test přesného porovnání, který porovnával upravený tvar s původním tvarem v referenční množině, a jednak test porovnání lemmat. Graf na obrázku 6.5 ukazuje rozdíl hodnot úplnosti pro tyto testy provedené na výsledcích



Obrázek 6.5: Graf úplnosti v závislosti na množství vybraných klíčových slov z českých norem bez úpravy tvarů.

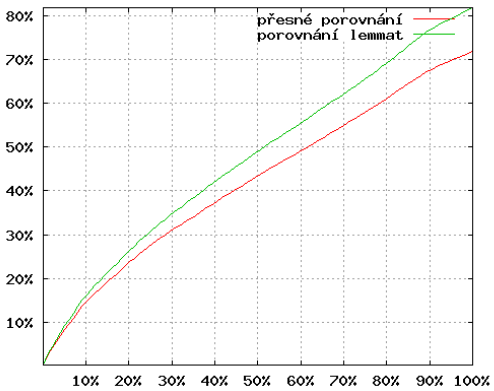


Obrázek 6.6: Graf úplnosti v závislosti na množství vybraných klíčových slov z českých norem po úpravě tvarů.

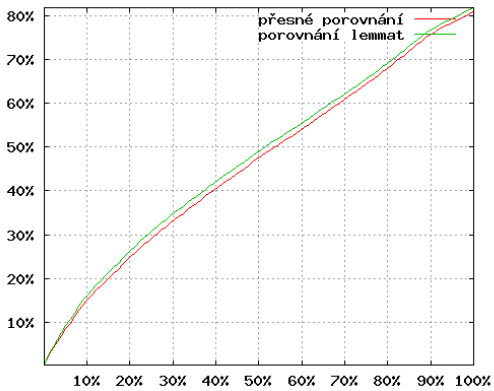
bez úpravy tvarů. Z grafu vyplývá, že až 31 % správně nalezených klíčových slov bylo v odlišném tvaru, než je žádoucí. Stejné testy na obrázku 6.6 provedené na upravených klíčových slovech jasně ukazují zlepšení, které je patrné z přiblížení obou křivek k sobě. Pouze 3,5 % klíčových slov bylo v jiném tvaru, než obsahovala referenční množina. To se dá považovat za úspěšný výsledek.

6.4.2 Anglická klíčová slova

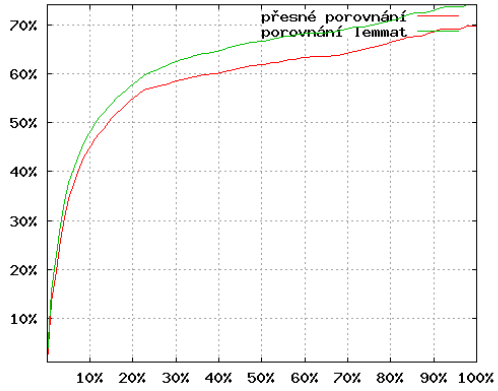
Úprava anglických slov probíhala pouze u těch výrazů, které se v textu vyskytly jen v jednom tvaru. V anglických normách bylo provedena úprava, která měla vliv na změnu výsledného tvaru, u 5,5 % vyhledaných klíčových slov. V článcích bylo takto upraveno dokonce 25 % výsledků.



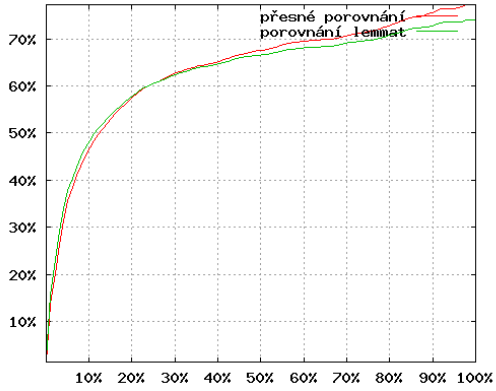
Obrázek 6.7: Graf úplnosti v závislosti na množství vybraných klíčových slov z anglických norem bez úpravy tvarů.



Obrázek 6.8: Graf úplnosti v závislosti na množství vybraných klíčových slov z anglických norem po úpravě tvarů.



Obrázek 6.9: Graf úplnosti v závislosti na množství vybraných klíčových slov z anglických článků bez úpravy tvarů.



Obrázek 6.10: Graf úplnosti v závislosti na množství vybraných klíčových slov z anglických článků po úpravě tvarů.

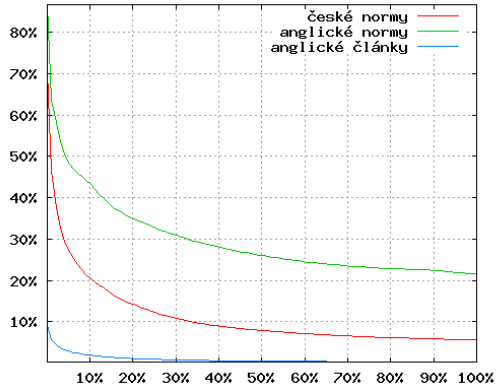
Z grafů úplnosti před úpravou (obrázky 6.7 a 6.9) je patrné, že v anglickém jazyce se opravdu nejedná o tak závažný problém. Tvary byly nesprávně určeny pro 10 % klíčových slov v normách a 4,5 % klíčových slov v článcích. I přesto je na grafů vytvořených se zapojenou úpravou tvarů slov jasně vidět zlepšení (obrázky 6.8 a 6.10). Počet chyb poklesl u norem na 1 %. U článků dokonce došlo ke zvýšení úplnosti o 3,3 %. To je na první pohled zarážející výsledek. Při bližším prozkoumání se však všechno vysvětlilo. Výběrem tvarů z textu se částečně eliminovaly některé chyby, kterých se dopustil analyzátor při tvorbě lemmat. Ukázalo se tedy, že i takto jednoduchý pokus o úpravu vyhledaných klíčových slov byl velmi úspěšný.

6.5 Předúprava vstupních textů

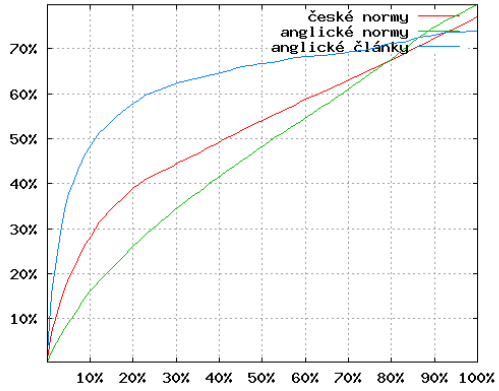
Při měření přínosu filtrace textu byly na předupravených dokumentech provedeny stejné testy jako v části 6.2. Přestože filtrace byla velmi úspěšná a byly odstraněny jak všechny samostatně stojící vzorce, tak všechny spojovníky, výsledky testů se nijak výrazně nezměnily. Z grafu 6.11 není pouhým okem pozorovatelná žádná změna. Při pečlivé kontrole výstupů zjistíme, že došlo k mírnému zvýšení přesnosti, ne však více než než o 1 %.

Při bližším prozkoumání byla odhalena příčina tak nevalného zlepšení. Vzorce se ve zkoumaných dokumentech neobjevovaly příliš často. Normy neobsahovaly žádné do textu převedené vzorce a v anglických článcích se objevily v minimální míře. Prakticky celé zlepšení připisuji odstraňování spojovníků z konce řádků. Vzhledem k tomu, že pravděpodobnost výskytu rozděleného výrazu, který by měl být klíčovým slovem, je malá, není ani zlepšení příliš vysoké.

Použití předúpravy dokumentů je tedy na zvážení. Vzhledem k tomu, že vyhledávání žádným způsobem nezhoršila a že je možné ji provést v krátkém čase, takže nebrzdí ostatní výpočty, rozhodl jsem se ji zahrnout do výsledné konfigurace systému. Filtrace textů byla vytvořena jako samostatný nástroj, takže je možné ji jednoduše zapojit nebo vynechat v závislosti na charakteru dokumentů zpracovávaných vyhledávačem.



Obrázek 6.11: Graf přesnosti v závislosti na množství vybraných klíčových slov. Měřeno na filtrovaných vstupních textech.



Obrázek 6.12: Graf úplnosti v závislosti na množství vybraných klíčových slov. Měřeno na filtrovaných vstupních textech.

6.6 Výsledná konfigurace systému

Do výsledné konfigurace systému byla zahrnuta předúprava dokumentů, vyhledání kandidátů na základě definovaných vzorů, ohodnocení kombinací metod *BM25*, *Lexical Cohesion*, *TF-IPF* a *TF-IDF* a úprava do výsledných tvarů. Vyhledávání kandidátů bylo navíc rozšířeno o seznam stop slov tvořeného regulárními výrazy, které akceptují pouze slova delší než 1, která obsahují jen písmena a případně jednu pomlčku uvnitř slova.

Pro odhalení všech správně vyhledaných výrazů byl použit test zohledňující různé uspořádání slov. Ten porovnává lemmata a přijme jenom jejich smysluplné kombinace. Na základě vyhledávacích vzorů bylo vytvořeno následující nastavení pravidel pro vytváření obměn klíčových slov (tabulka 6.8).

vzor v ref. množině	seznam akceptovaných vzorů
AN	AN, NA
AN_1N_2	AN_1N_2, N_1N_2A
N_1AN_2	N_1AN_2, N_1N_2A
DAN	DAN, NAD
A_1A_2N	A_1A_2N, NA_1A_2, NA_2A_1, A_1NA_2, A_2NA_1, A_2A_1N
NV	NV, VN
VN	VN, NV

Tabulka 6.8: Nastavení testu zohledňujícího různé pořadí slov.

V grafech 6.13 a 6.14 vynesených pro testy na českých normách je vidět, že všechny tři křivky leží poměrně blízko sebe. Z toho lze vyvozovat, že úprava do výsledných tvarů dává velmi dobré výsledky, protože se křivka vytvořená testem s přesným porovnáním blíží křivce vzniklé porovnáváním lemmat. Tato skutečnost už však byla ukázána dříve. Daleko důležitější je, že se příliš nevzdalují ani hodnoty naměřené kontrolou různého uspořádání slov. To vypovídá o správně určeném pořadí slov ve víceslovných výrazech. Přihlédneme-li k tomu, že je tvořeno pouze z informací o struktuře větného stromu, jedná se o ne zcela

triviální úkol. Řazení slov v anglických výrazech tolik obtížné nebylo, protože vychází přímo z původního pořadí slov v textu.

Použití seznamu stop slov a filtrace mělo vliv na zvýšení přesnosti vyhledávání. V českých dokumentech pro 10 % vybraných klíčových slov vzrostla přesnost o 3,1 %. V anglických normách na stejné úrovni vzrostla přesnost o 2 %, v článcích jen o 1 %.

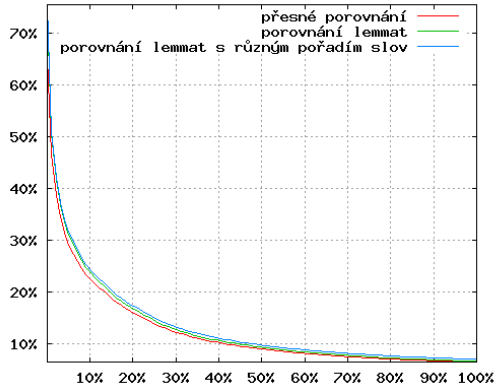
soubor dokumentů	porovnání s ref. množinu	subjektivně vhodná kl. slova
české normy	67,0 %	71,0 %
anglické normy	65,0 %	73,0 %
anglické články	9,0 %	34,0 %

Tabulka 6.9: Přesnosti pro prvních 300 vybraných výrazů měřené přesným porovnáním s referenční množinou a se započítanými slovy, která byla shledána vhodnými.

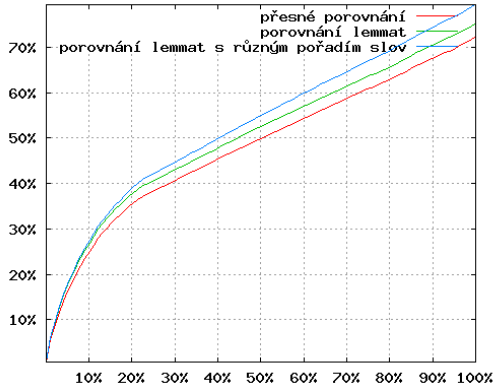
Závěrečná konfigurace systému byla podrobena také detailnímu zkoumání, kdy bylo prvních 300 vybraných výrazů z každé sady dokumentů ručně kontrolováno. Do výsledků byla započítána i ta klíčová slova, která se sice nenacházela v referenční množině, ale byla shledána vhodnými. Výsledek tohoto testu je pochopitelně ovlivněn subjektivním úsudkem.

Ve všech výstupech byla, jak se dalo předpokládat, nalezena ještě další vhodná klíčová slova. Především u článků, které obsahují minimum vlastních klíčových slov, však přesnost vzrostla. Nárůst činí 25 %, což není zanedbatelné. I tak je ale celková přesnost poměrně nízká a zcela jistě je zde prostor pro zlepšování.

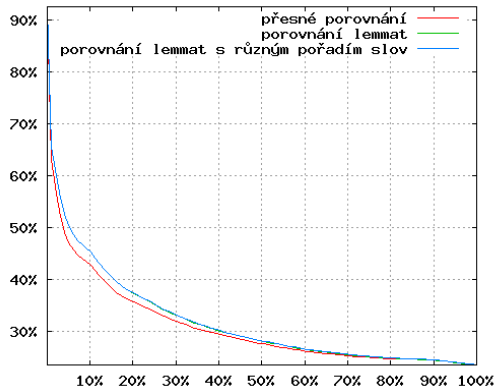
Z vynesených grafů úplnosti vyplývá, že pro všechny testované soubory dat se určitě vyplatí vybrat prvních 20 % vyhledaných klíčových slov. Za touto úrovní se všechny grafy ať už více či méně znatelně lámou a je vybíráno méně vhodných výrazů.



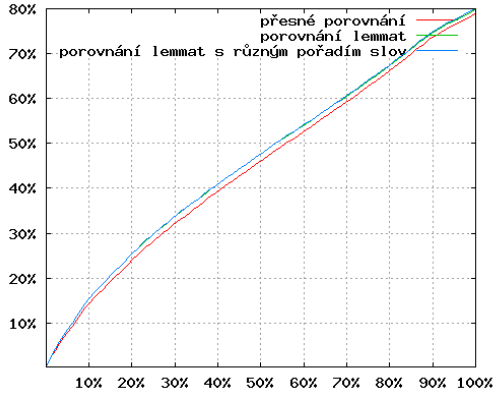
Obrázek 6.13: Graf úplnosti v závislosti na množství vybraných klíčových slov z českých norem.



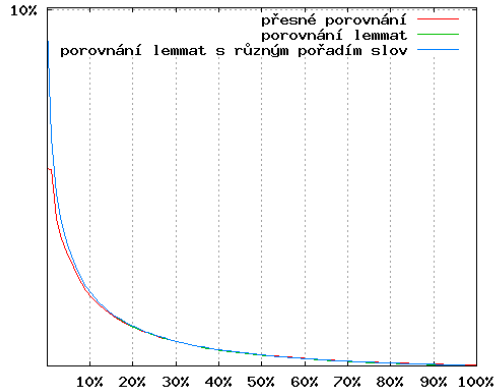
Obrázek 6.14: Graf přesnosti v závislosti na množství vybraných klíčových slov z českých norem.



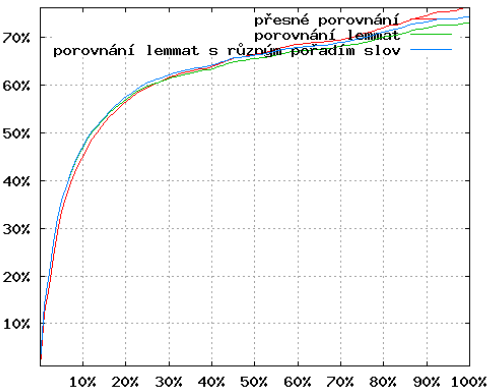
Obrázek 6.15: Graf úplnosti v závislosti na množství vybraných klíčových slov z anglických norem.



Obrázek 6.16: Graf přesnosti v závislosti na množství vybraných klíčových slov z anglických norem.



Obrázek 6.17: Graf přesnosti v závislosti na množství vybraných klíčových slov z anglických článků.



Obrázek 6.18: Graf úplnosti v závislosti na množství vybraných klíčových slov z anglických článků.

Kapitola 7

Závěr

Cílem předkládané práce bylo vytvořit systém pro navrhování klíčových slov. Za účelem tvorby takového systému byl navržen a implementován framework, který umožňuje zpracovávat dokumenty označkované lingvistickými analyzátory. Tento framework je dostatečně obecný a umožňuje stavět nástroje pro různé jazyky. Konkrétně bylo vytvořeno rozhraní pro práci s výstupy analyzátorů *TreeTagger*, *MiniPar* a *PDT 2.0*. Část frameworku určená pro manipulaci s obsahem dokumentů je využitelná i pro jiné úkoly z oboru zpracování přirozeného jazyka. Dál bylo implementováno vyhledávání kandidátů, kteří jsou následně ohodnoceni různými algoritmy.

Kandidáti na klíčová slova jsou vybíráni na základě vzorů, které umožňují přihlédnout k závislostem ve větných stromech. Součástí vzorů je i určování pozice slov ve víceslovných výrazech. Byly vybrány vyhledávací vzory, které pokrývají zhruba 80 % referenčních klíčových slov. Nastavení pořadí slov se v testech ukázalo jako vhodné.

Pro vyhledávání klíčových slov bylo implementováno několik algoritmů, z nichž jen čtyři byly na základě jejich výsledků vybráni do výsledné konfigurace systému. Všechny tyto čtyři metody využívají pro výpočet ohodnocení pouze četnosti a statistiky jednoduše vypočítatelné z dokumentů, což zaručuje vysokou rychlost zpracování. Pro kombinování výsledků algoritmu byla použita metoda *Weighted Voting*. Tato konfigurace dokázala při zachování dostatečné přesnosti v testech uspořádat vhodná klíčová slova v horních 20 % všech vybraných výrazů. Méně dobrých výsledků bylo dosaženo na článcích z anglických konferencí. Zde je jistě ještě prostor pro další vylepšení. Vhodnou oblastí k prozkoumání by mohlo být vytvoření referenčního korpusu ze všech dostupných článků a nalezení vhodné hodnotící metody, která by s ním pracovala.

Vybraná klíčová slova byla upravována do vhodných tvarů. To má velký význam především v českých textech. Navržené úpravy probíhají ve dvou krocích. Nejprve jsou vyhledávány správné tvary přímo v textu. Pokud tento pokus selže, je tvar určen podle nastavených pravidel úpravy. Testy ukázaly, že zvoleným postupem lze výběr vhodných tvarů zlepšit až o 31 %. To se dá, myslím, považovat za velmi dobrý výsledek. Úprava anglických slov byla realizována výrazně jednodušším způsobem. Přesto však byla v testech prokázána zlepšení, které přinesla.

Součástí práce bylo i prověření možností použití slovníku synonym pro sjednocení klíčových slov. Navržený postup se však v praxi neosvědčil a neměl pozitivní vliv na dosažené výsledky. Proto byl ze závěrečné konfigurace systému vypuštěn. Další zkoumanou oblastí byla filtrace vstupních textů za účelem odstranění těch částí, které způsobují chyby při lingvistické analýze. Vytvořený nástroj zvládá svůj úkol velmi dobře. Vzhledem k vláknové implementaci navíc pracuje rychle. V této práci se díky charakteru zpracovávaných doku-

mentů podařil prokázat pouze mírný vliv takovéto úpravy na zlepšení výsledků. Přesto však má dle mého názoru filtrace místo v dalším použití například na jiném typu dokumentů.

Přínos této práce vidím především ve vytvoření základu pro další zkoumání a rozvoj, a to nejen v oblasti vyhledávání klíčových slov, ale i v oboru zpracování přirozeného jazyka jako celku. Systém je otevřen pro jakákoliv budoucí rozšíření. Na základě provedených testů byla sestavena konfigurace sytému, která se osvědčila hlavně pro vyhledávání klíčových slov v normách. Další, jistě také přínosnou částí, bylo prověření úpravy výsledných tvarů, které se v testech velmi osvědčilo.

Literatura

[1] Okapi BM25. `http://en.wikipedia.org/wiki/Okapi_BM25`, 2009 [cit. 2009-12-26].

[2] About OpenOffice.org. `http://about.openoffice.org/`, 2010 [cit. 2010-05-12].

[3] Stopslovo. `http://cs.wikipedia.org/wiki/Stopslovo`, 2010 [cit. 2010-05-13].

[4] Aoe, J.; Morimoto, K.; Sato, T.: An efficient implementation of trie structures. *Software - Practice and Experience*, ročník 22, č. 9, 1992: s. 695–721.

[5] Church, K.: A stochastic parts program and noun phrase parser for unrestricted text. In *Proc. of ANLP*, 1988.

[6] Frantzi, K.; Ananiadou, S.; Mima, H.: Automatic recognition of multi-word terms:. the C-value/NC-value method. *International Journal on Digital Libraries*, ročník 3, č. 2, 2000: s. 115–130.

[7] Freund, Y.; Schapire, R.; Abe, N.: A short introduction to boosting. *JOURNAL-JAPANESE SOCIETY FOR ARTIFICIAL INTELLIGENCE*, ročník 14, 1999: s. 771–780.

[8] Hajič, J.; Hajičová, E.; Rosen, A.: Formal Representation of Language Structures. *TELRI Newsletter*, , č. 3, 1996: s. 12–19.

[9] Hajič, J.: Czech „Free“ Morphology. `http://ufal.mff.cuni.cz/pdt/Morphology_and_Tagging/Morphology/`, 2001 [cit. 2010-05-09].

[10] Hajič, J.; Hajičová, E.; Hlaváčová, J.; aj.: Průvodce PDT 2.0. 2006.

[11] Hlavsa, Z.; Martincová, O.: *Pravidla českého pravopisu*. Pansofia, 1993.

[12] Kempe, A.: A probabilistic tagger and an analysis of tagging errors. *Research Report. IMS, Univ. of Stuttgart*, 1994.

[13] Knoth, P.; Schmidt, M.; Smrz, P.; aj.: Towards a Framework for Comparing Automatic Term Recognition Methods. 2009.

[14] Korkontzelos, I.; Klapaftis, I.; Manandhar, S.: Reviewing and evaluating automatic term recognition techniques. *Lecture Notes in Computer Science*, ročník 5221, 2008: s. 248–259.

[15] Lauriston, A.: Criteria for measuring term recognition. In *Seventh Conference of the European Chapter of the Association for Computational Linguistics*, 1995, s. 27–31.

[16] Lin, D.: Principle-based parsing without overgeneration. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, ročník 31, Association for Computational Linguistics, 1993, s. 112–112.

[17] Lin, D.: Principar-an efficient, broad-coverage, principle-based parser. In *Proceedings of COLING*, ročník 94, 1994, s. 42–48.

[18] Lin, D.: Dependency-based evaluation of MINIPAR. *Treebanks: building and using parsed corpora*, 2003.

[19] Manning, C.; Schütze, H.: *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999, ISBN 0-262-13360-1.

[20] Marcus, M.; Santorini, B.; Marcinkiewicz, M.: Building a large annotated corpus of English: The Penn Treebank. *Computational linguistics*, ročník 19, č. 2, 1994: s. 313–330.

[21] Mašláňová, M.: *Automatická identifikace klíčových slov*. Diplomová práce, Vysoké učení technické v Brně, Fakulta informačních technologií, 2007.

[22] Mikulová, M.; Bémová, A.; Hajič, J.; aj.: Anotace Pražského závislostního korpusu na tektogramatické rovině: pokyny pro anotátory. Technická zpráva, ÚFAL MFF UK, Prague, Czech Republic, 2005.

[23] Miller, G.; Beckwith, R.; Fellbaum, C.; aj.: Introduction to wordnet: An on-line lexical database*. *International Journal of lexicography*, ročník 3, č. 4, 1990: str. 235.

[24] Miller, G. A.: WordNet - About Us. `http://wordnet.princeton.edu`, 2009 [cit. 2010-05-05].

[25] Čsn norma: *Čsn iso 999-1998 zásady zpracování uspořádání a grafické upravy rejstříků*. 1998.

[26] Park, Y.; Byrd, R.; Boguraev, B.: Automatic glossary extraction: beyond terminology identification. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, Association for Computational Linguistics Morristown, NJ, USA, 2002, s. 1–7.

[27] Patry, A.; Langlais, P.: Corpus-based terminology extraction. In *Proceedings of the 7th International Conference on Terminology and Knowledge Engineering*, 2005, s. 313–321.

[28] Česlav Przywara: *Metody extrakce víceslovných výrazů z textu*. Diplomová práce, Univerzita Karlova v Praze, Matematicko-fyzikální fakulta, 2008.

[29] Robertson, S.; Walker, S.: Okapi/keenbow at trec-8. *NIST SPECIAL PUBLICATION SP*, 2000: s. 151–162.

[30] Sampson, G.: The SUSANNE corpus. *ICAME Journal*, ročník 17, 1993: s. 125–127.

[31] Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, ročník 12, Manchester, UK, 1994.

[32] Sclano, F.; Velardi, P.: Termextractor: a web application to learn the shared terminology of emergent web communities. In *Proc. of the 3rd International Conference on Interoperability for Enterprise Software and Applications I-ESA*, Springer, 2007, s. 28–30.

[33] Wong, W.; Liu, W.; Bennamoun, M.: Determining termhood for learning domain ontologies using domain prevalence and tendency. In *Proceedings of the sixth Australasian conference on Data mining and analytics-Volume 70*, Australian Computer Society, Inc., 2007, s. 47–54.

[34] Zeman, D.; Hana, J.; Hanová, H.; aj.: A Manual for Morphological Annotation, 2nd edition. Technická Zpráva 27, ÚFAL MFF UK, Prague, Czech Republic, 2005.

[35] Zhang, Z.; Iria, J.; Brewster, C.; aj.: A comparative evaluation of term recognition algorithms. In *Proceedings of the sixth international conference of Language Resources and Evaluation (LREC 2008)*, 2008.

Dodatek A

Obsah CD

- Technická zpráva ve formátu PDF.
- Zdrojový text technické zprávy pro LaTeX.
- Zdrojové kódy knihoven pro práci s označkovanými soubory a vyhledávání klíčových slov.
- Definice vyhledávacích vzorů, pravidla pro úpravu do výsledných tvarů a použité četnosti slov.